# Proceedings of the International Multiconference on Computer Science and Information Technology

Volume 3 (2008)

**Proceedings of the International Multiconference on Computer Science and Information Technology**

**Volume 3 (2008)**

M. Ganzha, M. Paprzycki, T. Pełech-Pilichowski (editors)

ISSN 1896-7094

Polskie Towarzystwo Informatyczne
Oddział Górnośląski
ul. Raciborska 3
40-074 Katowice

ISBN 978-83-60810-14-9

IEEE Computer Society Press
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
USA

TEXnical editor: Alexander Denisjuk

# Proceedings of the International Multiconference on Computer Science and Information Technology

October 20 – 22, 2008. Wisła, Poland

Volume 3 (2008)

DEAR Reader, it is our pleasure to present to you Proceedings of the 2008 International Multiconference on Computer Science and Information Technology (IMCSIT'08), which will take place in Wisła, Poland, on October 20-22, 2008. IMCSIT'2008 will be co-located with the XXIV Autumn Meeting of the Polish Information Processing Society (PIPS).

IMCSIT is a result of the evolutionary process. In 2005 a Scientific Session took place during the XXI Autumn Meeting of PIPS and consisted of 27 refereed presentations. After this relative success (we have advertised the Session very late in the year) we have decided to expand and extend it into a full-blown conference but continue cooperation (co-location) with the Autumn Meetings of PIPS. As a result of a steady growth, in 2008, IMCSIT will consists of the following events (and Proceedings are organized into sections that correspond to each of them):

- Workshop on Agent Based Computing (ABC V),
- 3rd International Symposium: Advances in Artificial Intelligence and Applications (AAIA'07),
- Workshop on Computer Aspects of Numerical Algorithms (CANA'08),
- Computational Linguistics—Applications (CLA'08),
- 8th International Multidisciplinary Conference on e-Commerce and e-Government (ECOM-08),
- 4th Workshop on Large Scale Computations on Grids (LaSCoG'08),
- First International Symposium Multimedia—Applications and Processing (MMAP'08),
- International Conference on Principles of Information Technology and Applications (PITA'08),
- International Workshop on Real-Time Software (RTS'08),
- 3rd International Workshop on Secure Information Systems (SIS'08),
- Workshop on Ad-Hoc Wireless Networks: Urban Legends and Reality (WAHOC'08),
- Workshop on Computational Optimization (WCO'08).

These conferences had their own Organizers and a Program Committees (listed in these Proceedings). We would like to express our warmest gratitude to all of them for their hard work in attracting and later refereeing 214 submissions.

IMCSIT'2008 was organized under patronage of Professor Barbara Kudrycka, Minister of Science and Higher Education and Waldemar Pawlak, Deputy Prime Minister and Minister of Economy. Furthermore, IMCSIT'2008 was organized in cooperation with the Poland Chapter of the IEEE Computer Society, Systems Research Institute Polish Academy of Sciences, Institute of Computer Science of the Polish Academy of Sciences, Institute of Innovation and Information Society, and Asociación de Técnicos de Informática.

Additionally, the 3rd International Symposium: Advances in Artificial Intelligence and Applications (AAIA'08) was devoted to the memory of Professor Ryszard S. Michalski, and organized in cooperation with:

- Poland Chapter of the IEEE Computational Intelligence Society,
- Polish Neural Network Society,
- World Federation of Soft Computing,
- European Society for Fuzzy Logic And Technology, and
- European Neural Network Society.

We would also like to mention that the 1st International Symposium: Advances in Artificial Intelligence and Applications (AAIA'2006) was devoted to the memory of Professor Zdzisław Pawlak. During this event a special awards carrying his name and recognizing "a general contribution" and "a contribution by a young researcher" were awarded. This year the Zdzislaw Pawlak Awards will be presented for the third time. They are sponsored by the Mazowsze Section and the Board of the Polish Information Processing Society.

Finally, we have utilized a unique opportunity brought about by co-location of a scientific/academic event (IMCSIT'2008) and a business IT oriented one (Autumn Meeting of PIPS) to address the inter-relations between the two. To this effect, in cooperation with the NESSI Technology Platform, we have organized a special joint session devoted to the IT Research and Development activities in Poland and in the European Union. During this session, for the first time, a Jan Łukasiewicz award will be given to the most innovative IT company in Poland.

During IMCSIT'2008 we plan three keynote presentations: Paolo Bresciani will talk about "Software and Services in the Future Internet; the 7th Framework Programme Perspective," Mike Hinchey will present a talk entitled "You Can't Get There from Here!Problems and Potential Solutions in Developing New Classes of Complex Computing Systems," finally, Zdravko Karakehayov, who is an IEEE Distinguished Visitor will present a talk devoted to "Wireless Ad-hoc Networks: Where Security, Real-Time and Lifetime Meet." Papers based on the latter two presentations can be found in these Proceedings.

We would like to express our gratitude to the Minister of Science and Higher Education for financial support in preparing and publishing this volume.

We hope that you will enjoy this volume and find it useful in your work. We would also like to invite you to participate in the next edition of IMCSIT as well Autumn Meeting of PIPS which will take place in Mrągowo in October of next year.

MEMBERS OF THE 2008 IMCSIT ORGANIZING COMMITTEE:

**Maria Ganzha,** Conference Chair, Systems Research Institute Polish Academy of Sciences, Warsaw and Elbląg University of Humanities and Economy, Elbląg.

**Marcin Paprzycki,** Systems Research Institute Polish Academy of Sciences, Warsaw and Management Academy, Warsaw.

**Tomasz Pełech-Pilichowski,** AGH University of Science and Technology, Kraków.

# Proceedings of the International Multiconference on Computer Science and Information Technology

Volume 3

**October 20 – 22, 2008. Wisła, Poland**

## TABLE OF CONTENTS

### Workshop on Agent Based Computing V:

### 3rd International Symposium Advances in Artificial Intelligence and Applications:

## Computer Aspects of Numerical Algorithms:

## Computational Linguistics–Applications:

## 8th International Multidisciplinary Conference on e-Commerce and e-Government:

## $4^{\text{th}}$ Workshop on Large Scale Computations on Grids:

## First International Symposium on Multimedia–Applications and Processing:

## International Conference on Principles of Information Technology and Applications:

## International Workshop on Real Time Software:

## 3$^{\text{rd}}$ International Workshop on Secure Information Systems:

## Workshop on Wireless and Unstructured Networking: Taming the Stray:

## Workshop on Computational Optimization:

# Workshop on Agent Based Computing V

ORGANIZED within the framework of the International Multiconference on Computer Science and Information Systems and co-located with the XXIV Autumn Meeting of the Polish Information Processing Society.

Agent-based computing has been hailed as the next significant breakthrough in software development, with the potential to affect many aspects of computer science, from artificial intelligence to the technologies and models of distributed computation. Agent-based systems are capable of autonomous action in open, dynamically-changing environments. Agents are currently being applied in domains as diverse as business information systems, computer games and interactive cinema, information retrieval and filtering, user interface design and industrial process control. The aim of the ABC'08 is to bring together researchers and developers from industry and academia in order to report on the latest scientific and technical advances, discuss and debate the major issues, and showcase the latest systems.

ABC Workshop welcomes submissions of original papers concerning all aspects of software agents. Submissions involving both implementation and theoretical issues are encouraged.

Topics include but are not limited to:

- Agent architectures
- Agent-oriented software engineering
- Agent-based simulations
- Agent benchmarking and performances
- Agent communication, coordination and cooperation
- Agent languages
- Agent learning and planning
- Agent mobility
- Agent modeling, calculi, and logics
- Agent security
- Agents and Service Oriented Computing
- Agents in the Semantic Web
- Applications and Experiences
- Simulating and verifying agent systems
- Multiagent Systems Product Lines
- Multi-Agent Systems in Applications
- Disaster Management
- E-commerce
- Workflow management and business processes
- Other areas

## Organizing Committee

**Maria Ganzha,** EUH-E and IBS PAN, Poland
**Marcin Paprzycki,** WSM and IBS PAN, Poland
**Shahram Rahimi,** Southern Illinois University, USA

## International Programme Committee

**Myriam Abramson,** Naval Research Laboratory, USA

**Stanislaw Ambroszkiewicz,** Institute of Computer Science, Polish Academy of Sciences, Poland

**Costin Badica,** University of Craiova, Romania

**Giacomo Cabri**, Università di Modena e Reggio Emilia, Italy

**Radovan Cervenka,** Whitestein Technologies AG, Slovakia

**Krzysztof Cetnarowicz,** AGH - University of Science and Technology, Poland

**Walter Binder,** University of Lugano, Switzerland

**Bengt Carlsson,** Blekinge Institute of Technology, Sweden

**Vladimir Gorodetsky,** SPII RAS, Russia

**Dominic Greenwood,** Whitestein Technologies AG, Switzerland

**Henry Hexmoor,** Southern Illinois University, USA

**Tomas Klos,** Delft University of Technology, The Netherlands

**Zofia Kruczkiewicz,** Wroclaw University of Technology, Poland

**Michele Loreti,** Università di Firenze, Italy

**Beniamino di Martino,** Seconda Università di Napoli, Italy

**Viorel Negru,** Western University of Timisoara, Romania

**Ngoc-Thanh Nguyen,** Wroclaw University of Technology, Poland

**Benno Overeinder,** Vrije Universiteit Amsterdam, The Netherlands

**Myon Woong Park,** Korea Institute of Science and Technology, Korea

**Michal Pechouczek,** Czech Technical University, Czech Republic

**Joaquin Pena,** University of Seville, Spain

**Volker Roth,** FX Palo Alto Laboratory, Inc. , USA

**Jarogniew Rykowski,** Poznan University of Economics, Poland

**Sattar B. Sadkhan,** University of Babylon, Iraq

**Stanislaw Stanek,** Katowice University of Economics, Poland

**Paolo Torroni,** University of Bologna, Italy

**Walt Truszkowski**, NASA—Goddard Space Flight Center, USA

**Rainer Unland,** University of Duisburg-Essen, Germany

**Andrzej Uszok,** Florida Institute for Humane and Machine Cognition, USA

**Tatyana Yakhno,** Dokuz Eylul University, Turkey

**Yu Zhang,** Trinity University, USA

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Agent architecture for building Robocode players with SWI-Prolog

Vasile Alaiba, Armand Rotaru
Al.I.Cuza University,
Department of Computer Science
General Berthelot 16
Iaşi, România
Email: {alaiba, armand.rotaru}@info.uaic.ro

***Abstract*—Effective study of autonomous agents is a challenging activity, for researchers and programmers alike. In this paper we introduce a new agent environment built on top of the tank-simulation game Robocode. We consider this game a good choice for testing agent programming strategies and an excellent learning opportunity. We integrated SWI-Prolog into Robocode and built an architecture for writing agents with Prolog. Game generated events are translated by our framework into predicate invocations, allowing the agent to perceive the environment. An API to issue commands to the tank from Prolog acts as agent's effectors. We tested our architecture by implementing a simple autonomous agent that uses a static strategy.**

## I. Motivation

AGENT oriented programming is hailed to be the next paradigm shift in software engineering. Four key notions were identified that distinguish agents from arbitrary programs: reaction to the environment, autonomy, goal-orientation and persistence [4]. A lot of research goes in the areas connected to the field. Almost unquestionable goes the statement that the most important feature is autonomy, without which an agent can not be. A system is autonomous to the extent that its behavior is determined by its own experience [5].

The agent related ideas were around for some time and under different names. If we are to consider programming games like Robocode [3], we will see that these robots not only fulfill most if not all of the requirements to be named "agents", but the environment itself is well defined and complex enough to provide insight into harder real-life problems. We will describe with greater detail later on the characteristics of the game and the challenges it poses.

Logic programming, both as a field of science and as a practical tool, is old and new at the same time. Starting with the famous statement made by Kowalski, "algorithm = logic + control", made back in 1979 [6], the field knew ups and downs for the last 30 years. Although this may seem relatively old for the computer science field, there is still a lot of room for improvement. Software agents, especially, can benefit from a declarative logic programming language like Prolog and its dialects.

Last but not least, we strongly believe that research should be done as close to a practical field as possible, in order to benefit from a quick feedback loop. We chose the tank-simulation game Robocode to be our practical software virtual environment and Prolog our agent programming language. Thus we provide a new architecture for developing and experimenting agent programming techniques with ease.

## II. Robocode: the agent's environment

Robocode is a robotics battle simulator that runs across all platforms supporting any recent Java runtime (version 2 or more). It is an easy-to-use framework that was originally created to help teach object-oriented programming in Java [1]. The project was started by M. Nelson in late 2000 and was available through IBM's *alphaWorks* in July 2001. Now there is a strong open-source community around it with competitions runing all over the world [3].

The basic idea behind Robocode is letting autonomous software agents (robots) compete in a virtual closed environment against each other. Each robot is programmed in Java and has to implement a given base class. This code controls a virtual robot in arena by moving its body, steering, targetting and shooting other robots. The same base class provides handlers that the robot can implement to respond to external events. The whole system follows an event-driven architecture. Moreover, each robot runs in a security sandbox, allowing for easy and safe distribution of robot code throughout the community, protecting against harmful effects.

### A. Robocode simulation engine architecture

The Robocode game consists of several robots engaging in combat in a closed environment, a rectangular arena. The size of the arena and some other options can be configured just before a battle is started. As seen in figure 1, each robot runs on its own thread with its own event queue, responding to events generated by the game engine independently of the battle manager and the other robots [1]. In addition to this, each robot has a `run()` method where the main processing loop occurs. This is usually where the general strategy is implemented, i.e. what the robot does when no events occur.

Each robot is made up from three parts: the vehicle, the radar and the gun (figure 2). They can be moved both together and independent of each other. The vehicle can be moved ahead and back, in order to reposition the robot's body. Any of the parts can be turned left and right. Turning the vehicle changes the direction of further moves, turning the scanner

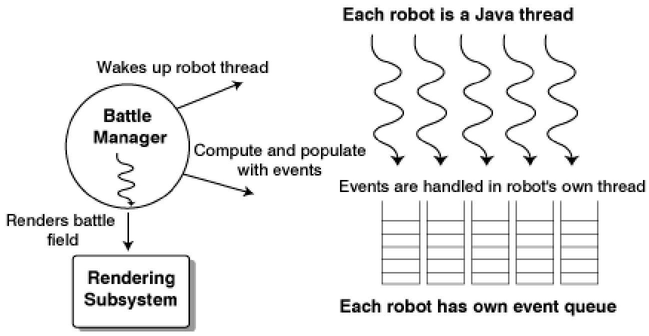

Each robot is a Java thread
Wakes up robot thread
Battle Manager
Compute and populate with events
Events are handled in robot's own thread
Renders battle field
Rendering Subsystem
Each robot has own event queue

Fig. 1. Robocode simulation engine architecture (see [1])

changes the area that the robot “senses” and turning the gun sets the direction where the bullet will be fired on.

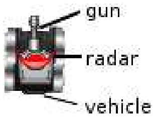


Fig. 2. Anatomy of a Robocode robot

## B. Robocode physics

Robocode *time* is measured in “ticks”, which are equivalent to frames displayed on the screen. Each robot gets one turn per tick. *Distance* measurement is done in pixels, with double precision, so a robot can actually move a fraction of a pixel. Robots *accelerate* at the rate of 1 pixel/tick and *decelerate* at the rate of 2 pixels/tick. Acceleration is automatically determined based on the distance the robot is trying to move. *Velocity* is calculated multiplying acceleration with time, and can never exceed 8. The direction is always the heading of the robot’s body.

Some limits are imposed for the rotation of a robot’s parts. Maximum rate of vehicle rotation is set such as it is limiting the ability to turn with speed. Maximum rate of gun rotation is 20 degrees per tick and is added to the current rate of rotation of the robot. Maximum rate of radar rotation is 45 degrees per tick and is added to the current rate of rotation of the gun.

Collisions cause damage, decreasing the energy level of the robot(s) involved. For example, hitting another robot damages both of them with 0.6, while a robot hitting a wall will take damage according to its speed.

# III. Logic programming agent architecture

The motivations behind our choice to implement a new logic programming agent architecture were already explained in section I. Still, we want to emphasize the practical need that drove our research effort. A strong community and momentum exists around Robocode both as an educational game and a tool for robot programming hobbists [3]. Agent oriented programming, on the other hand, is a field holding much promise not necessarily from a technological point of view, but mostly as a new design method. The shift in focus from designing parts (objects) that interact with each other in order to build a system that solves a problem to a more autonomous way of designing software poses new challanges. Artificial intelligence, and especially logics, play an important role in finding solutions to these problems. We believe that robot programming with Prolog is a field that was not explored enough. Our agent architecture is directed at making the entry into this field much easier for researchers and software engineers alike.

The architecture we propose is built around the natural event-driven nature of Robocode. We provide the means for a Prolog program to receive events from the game and to issue commands to it, thus implementing the basic requirements of an agent: “[it is] perceiving its environment through sensors and acting upon that environment through effectors” [5].

## A. Overall design

Any robot implementation in Robocode has to inherit from `robocode.Robot` class (or any subclass of it), and our design can not be an exception. We wrote a generic Java class `PrologRobotShared` that acts as an adaptor between the game environment and the concrete Prolog implementation of a robot. Its main responsability is to pass along events from the game engine, and to start the main strategy loop.

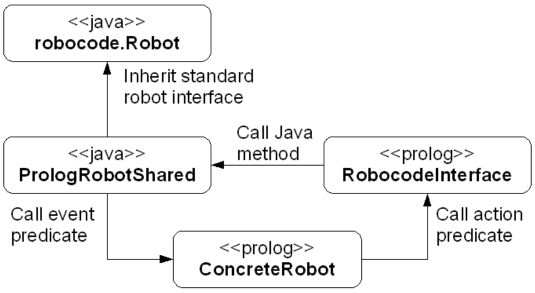


Fig. 3. Interaction model of a concrete Java-Prolog robot

By convention, the predicate that is being invoked when the game starts is `startStrategy`. Just as its Java counterpart, the `run` method, this predicate must not finish. Instead, this is where the overall strategy is specified. Acting upon the environment is possible using a collection of predicates that wrap the action calls from `robocode.Robot`. Calling any of these from the `startStrategy` predicate will freeze the robot thread while the game engine actually executes the command.

In order to improve the design and reusability of code between robot implementations, we separated any generic Prolog predicates to a different component (file), called `RobocodeInterface`. We also made sure that there is no Java code that needs to be changed if the robot’s logic is changed.

## B. Perceiving the environment

There are two ways of getting information regarding the environment. In the Robocode API there is a group of methods named `get*`. Most important ones are listed in table I. For all these wrapper predicates were implemented in `RobocodeInterface`, with exactly the same name. We can call this way of perceiving the environment *passive*, as these should be called from within the `startStrategy` predicate or any of the event handlers (see next section). The information they gather is always there, as it is the state of the game or of the robot.

The alternative way of perceiving the environment is by responding to events. A list of the most important events available is listed in table II. For all these a Java handler was written that calls the handler predicate with the same name from the concrete Prolog robot implementation, if it exists. We call this event-driven way *active*, as the environment interrupts the robot to inform it of a change in the environment or a sensory activity.

Maybe the most important of these methods is `onScannedRobot`. It is called when a robot's radar scan detects the presence of another robot. Responding to this event is usually a good strategy or at least a potentially strategy altering event.

### C. *Acting upon the environment*

A couple of "hardware" behavior properties can be altered using the `set*` methods listed in table III. These don't really count as effectors, but more like altering future behavior of effector methods. A couple more strictly cosmetic methods exist to set the appearance of the robot icon on the board. We did not implement these as Prolog predicates. If the user wants to change the default colors, they should do it directly in `PrologRobotShared`.

In table IV are listed the real effectors available to use for the robot's implementor. All these were implemented and are available in Prolog. Any call to these predicates will freeze the robot thread execution until the required action is completed, or an event occurs.

## IV. Reference implementation of an agent

Developing a framework can not be done without testing. We did a reference implementation of a robot in order to validate our approach, and provide an example of how to get started. We deliberately chose a simple robot, already present in the Robocode distribution as a sample. What we did was to start from the Java implementation and rewrite it using the framework. The end result is a functional Java-Prolog program implementing the same functional requirements as the original one.

### A. *Corners: a simple robot*

We chose a robot named "Corners" for our implementation. The strategy it applies is very simple: it chooses a corner, based on previous experiences, then proceeds to secure it. After it took the corner, it aims and shoots at the other robots using a fairly simple targetting algorithm.

### B. *Implementing the strategy in Prolog*

The strategy of a Robocode robot is made up, in general, of two parts: the main loop, triggered by the method `run()`, and the event handlers (see table II). Our implementation of the main loop initialises and customizes the robot, then calls the predicate `startStrategy` (listing 1). The first argument ($R$) of this predicate is received from Java and is a reference to the `PrologRobotShared` instance. It is necessary in order to issue commands back to the framework (actions, see table IV).

**Algorithm 1** Implementation of $startStrategy$ in Prolog

```
startStrategy(R) :-
        getOthers(R, Num),
        retractall(others),
        assert(others(Num)),
        goCorner(R),
        spinGunLoop(R, 3).

spinGunLoop(R, X) :-
        spinGun(R, X, 30),
        NewX is -X,
        spinGunLoop(R, NewX).
% ...
```

**Algorithm 2** Implementation of $onScannedRobot$ in Java and Prolog

```
public void
onScannedRobot(ScannedRobotEvent e) {
 executeQuery("onScannedRobot",
                new JRef(e));
}

onScannedRobot(R, E) :-
        stopWhenSeeRobot(Result),
        onScannedRobot(R, E, Result).

onScannedRobot(R, E, true) :-
        stop(R),
        getDistance(E, Distance),
        smartFire(R, Distance),
        scan(R),
        resume(R).

onScannedRobot(R, E, false) :-
        getDistance(E, Distance),
        smartFire(R, Distance).
%...
```

The event handlers are implemented in Java as delegators to Prolog, to predicates with similar names (listing 2). A ref-

TABLE I
ROBOCODE API: GATHERING INFORMATION ABOUT THE ENVIRONMENT AND SELF

| | | | |
|---|---|---|---|
| getBattleFieldHeight | the height of the current battlefield measured in pixels | getBattleFieldWidth | the width of the current battlefield measured in pixels |
| getGunCoolingRate | the rate at which the gun will cool down | getWidth | the width of the robot measured in pixels |
| getHeight | the height of the robot measured in pixels | getName | the robot's name |
| getNumRounds | the number of rounds in the current battle | getRoundNum | the current round number (0 to `getNumRounds` - 1) of the battle |
| getOthers | how many opponents are left in the current round | getTime | the game time of the current round |
| getGunHeading | the direction that the robot's gun is facing, in degrees | getGunHeat | the current heat of the gun |
| getEnergy | the robot's current energy | getHeading | the direction that the robot's body is facing, in degrees |
| getRadarHeading | the direction that the robot's radar is facing, in degrees | getVelocity | the velocity of the robot measured in pixels/turn |
| getX | the X position of the robot | getY | the Y position of the robot |

TABLE II
ROBOCODE API: EVENTS A ROBOT CAN RESPOND TO

| | | | |
|---|---|---|---|
| onScannedRobot | the robot sees another robot (i.e. the robot's radar scan "hits" another robot) | onStatus | called every turn in a battle round in order to provide the status to the robot |
| onBulletHit | called when one of robot's bullets hits another robot | onBulletHitBullet | called when one of robot's bullets hits another bullet |
| onBulletMissed | called when one of robot's bullets misses (i.e. hits a wall) | onHitByBullet | called when the robot is hit by a bullet |
| onHitRobot | called when the robot collides with another robot | onHitWall | called when the robot collides with a wall |
| onBattleEnded | called after end of the battle, even when the battle is aborted | onDeath | called if the robot dies |
| onRobotDeath | called when another robot dies | onWin | called if the robot wins a battle |

TABLE III
ROBOCODE API: BEHAVIORAL PROPERTIES THAT CAN BE SET

| | |
|---|---|
| setAdjustGunForRobotTurn | turn the gun independent from the robot's turn |
| setAdjustRadarForGunTurn | turn the radar independent from the gun's turn |
| setAdjustRadarForRobotTurn | turn the radar independent from the robot's turn. |

TABLE IV
ROBOCODE API: ACTIONS A ROBOT CAN TAKE

| | | | |
|---|---|---|---|
| ahead | immediately moves the robot forward | back | immediately moves the robot backward |
| doNothing | do nothing this turn (i.e. skip this turn) | stop | immediately stops all movement, and saves it for a call to `resume()` |
| resume | immediately resumes the movement you stopped by `stop()`, if any | turnLeft | immediately turns the robot's body to the left |
| turnRight | immediately turns the robot's body to the right | scan | scans for other robots |
| turnRadarLeft | immediately turns the robot's radar to the left | turnRadarRight | immediately turns the robot's radar to the right |
| fire/fireBullet | immediately fires a bullet | turnGunLeft | immediately turns the robot's gun to the left |
| turnGunRight | immediately turns the robot's gun to the right | | |

erence to the `PrologRobotShared` instance is passed to each predicate, to be able to call the action predicates.

## V. Related Work

Research on Agent architectures and frameworks is in progress for some time [9]. Even though, the field is still not developed enough and did not achieve the full embrace of industry yet. Logic programming is a good choice for implementing agents, as shown in several studies [10], [11].

The game Robocode was used in several studies before, including the creation of a team of robots [12] or a single robot's strategy [13]. Genetic programming was applied with some success, leading to generation of Robocode tank fighters [7], [8].

There is no study that we are aware of trying to implement Robocode fighters in Prolog.

## VI. Conclusion and Future Work

A new agent environment and framework for the creation of autonomous agents in Prolog was introduced. It is particularly interesting as it makes available the well known Robocode environment for programming in conjunction with the logic programming system SWI-Prolog. It also enables competitions with other agents built with different technologies, both created by humans and evolved [8]. It provides a good testing ground to validate different theories related to programming autonomous agents.

The framework can also be used to learn and experiment with agent-oriented programming, both in class and as a self-study tool. We plan to use it as learning environment for an undergraduate Logic programming course next year. The source code is released under an open-source GPL license and is available at http://www.info.uaic.ro/ alaiba/robocodepl/.

Several future directions can be followed. Our first priority is to extend the framework to support the full API of the latest version of Robocode (at the time of this writing 1.6.0). Tests will be done to determine the performance penalty, if it exists, that the framework adds on top of existing Java implementations of some common robots.

Programming teams of robots as multi-agent systems is another interesting area of research. Robocode supports the notion of teams and allows for collaborative combat [2]. Support in the framework should be added to enable inter-agent communicaton, both as point-to-point and as broadcast.

In the long term, we plan to add higher abstractions to the framework that will model concepts such as beliefs and goals, and support for different kinds of non-standard logics (such as temporal logic). Ideally this will lead to the development of design patterns for agent programming and their validation against a well defined, real environment.

## REFERENCES

[1] S. Li, “Rock ’em, sock ’em Robocode!, Learning Java programming is more fun than ever with this advanced robot battle simulation engine,” *IBM developerWorks,* 2002, http://www.ibm.com/developerworks/java/library/j-robocode/.

[2] S. Li, “Rock ’em, sock ’em Robocode: Round 2, Go beyond the basics with advanced robot building and team play,” *IBM developerWorks,* 2002, http://www.ibm.com/developerworks/java/library/j-robocode2/.

[3] *Robocode homepage,* http://robocode.sourceforge.net/.

[4] S. Franklin and A. C. Graesser, “Is it an agent, or just a program? A taxonomy for autonomous agents,” *in Intelligent agents,* iii, Springer Verlag, Berlin, 1997, pp. 21–35.

[5] J. S. Russell, P. Norvig, *Artificial intelligence: A modern approach,* Prentice Hall, Englewood Cliffs, NJ, 1995.

[6] R. A. Kowalski, “Algorithm = Logic + Control,” *in Comm. ACM,* 22(7), 1979, pp. 424–436.

[7] J. Eisenstein, “Evolving Robocode Tank Fighters,” *in CSAIL Technical Reports,* Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, 2003.

[8] Y. Shichel, E. Ziserman, “GP-Robocode: Using genetic programming to evolve robocode players,” *in Proceedings of 8th European Conference on Genetic Programming*, 2005.

[9] R. A. Brooks, “A robust layered control system for a mobile robot”, *in IEEE Journal of Robotics and Automation*, 2(1), 1986, pp. 14-23.

[10] J. A. Leite, J. J. Alferez, L. M. Pereira, “MINERVA - A Dynamic Logic Programming Agent Architecture”, *in Lecture Notes In Computer Science, Revised Papers from the 8th International Workshop on Intelligent Agents VIII*, 2333, Springer, 2001, pp. 141–157.

[11] S. Costantini, A. Tocchio, “The DALI Logic Programming Agent-Oriented Language”, *in Proceedings of 9th European Conference on Logics in Artificial Intelligence JELIA, Lecture Notes in Computer Science*, 3229, Springer, 2004, pp. 685-688.

[12] J. Frokjaer, P. B. Hansen, M. L. Kristiansen, I. V. S. Larsen, D. Malthesen, T. Oddershede, R. Suurland, “Robocode—Development of a Robocode team”, Technical note, Department of Computer Science, Aalborg University, 2004.

[13] K. Kobayashi, Y. Uchida, K. Watanabe, “A study of battle strategy for the Robocode”, *in Proceedings of SICE Annual Conference*, Fukui University, Japan, 2003.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Tackling Complexity of Distributed Systems: towards an Integration of Service-Oriented Computing and Agent-Oriented Programming

Giacomo Cabri, *Member IEEE*, Letizia Leonardi and Raffaele Quitadamo
Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia
Via Vignolese, 905 41100 Modena – Italy
Email: {giacomo.cabri, letizia.leonardi, raffaele.quitadamo}@unimore.it

***Abstract*—The development of distributed systems poses different issues that developers must carefully take into consideration. Web Services and (Mobile) Agents are two promising paradigms that are increasingly exploited in distributed systems design: they both try, albeit with very different conceptual abstractions, to govern unpredictability and complexity in wide-open distributed scenarios. In this paper, we compare the two approaches with regard to different aspects. Our aim is to provide developers with critical knowledge about the advantages of the two paradigms, stressing also the need for an intelligent integration of the two approaches.**

## I. Introduction

IN RECENT years, the interest in distributed computing has been ever-increasing both in industry and academia. Distributed computing offers advantages in its potential for improving availability and reliability through replication; performance through parallelism; sharing and interoperability through interconnection; flexibility and scalability through modularity. In order to gain these potential benefits, software engineers have been coping with new issues arising from distribution: components are scattered across network nodes and the control of the system is complicated by such a partitioned "system state", particularly challenging when dealing with failure and recovery; in addition, the interactions between the concurrent components give rise to issues of non-determinism, contention and synchronization. The key concept in distributed systems has been the *service*, implemented and provided by servers to clients that may dynamically join systems, locate and use required services and then depart. The last years' trend is toward the conception of services that can be globally accessible by remote clients over the Internet. Therefore, the technological and methodological background developed for conventional distributed systems often fail to scale up when applied to the design of large-scale systems. Languages for distributed programming were introduced so that each component could be described and used through an established *interface*, but language constructs turned out to be not enough. New paradigms were needed in order to tackle the growing *complexity* of software. Throughout this paper, we will discuss some of the critical aspects of modern distributed applications, showing where Agent-Oriented Programming (AOP) and Service-Oriented Computing (SOC) take different roads and how research efforts are being made to reconcile them.

## II. Background

The aim of distributed systems design is to identify the distributable components and their mutual interactions that together fulfil the system requirements. The client-server model is undoubtedly the most consolidated and applied paradigm in distributed computer system design. Pretty much every variation of application architecture that ever existed has an element of client-server interaction within it. Nevertheless, the last year trend has been in breaking up the monolithic client executable into object-oriented components (located part on the client, part on the server), trying to reduce the deployment headaches by centralizing a greater amount of logic on server-side components. The benefits, derived from the extensive use of the component-based approach, came at the cost of an increased complexity and ended up shifting ever more effort from deployment issues to maintenance and administration processes. The issues to tackle are not related only to how to partition the complex problem domain (i.e. the *problem space decomposition*) or where the identified components should reside (i.e. the *location awareness*), but an increasing emphasis is shifting on how these components should interact and should be maintained. As a consequence, researchers in software engineering are investigating the possibility to introduce paradigms born to deal with complexity even at the abstract model level. The two paradigms briefly discussed in the following subsections (i.e. *Service-Oriented Computing* and *Agent-Oriented Programming*) try to radically change the model entities upon which software designers use to build complex distributed systems. They introduced the powerful concepts of services and mobile software agents as the building blocks of the design, establishing then precise rules for their composition and interaction.

### A. Agent-Oriented Programming

An *agent* is basically defined as an entity that enjoys the properties of *autonomy*, *reactivity*, *proactivity*, *social abil-*

978-83-60810-14-9/08/$25.00 © 2008 IEEE

*ity* [27]. Software agents are significantly powerful when they live in communities made up of several interacting agents. Every agent is an active entity, situated in an environment, able to perceive and to react in a timely fashion to changes that occur in the environment. They are autonomous in the sense that they have the total control of their encapsulated state and are capable of taking decisions about what to do based on this state, without a third party intervention. In addition, agents exhibit goal-directed behaviour by proactively taking the initiative in pursuit of their design objectives.In an agent-oriented view, in order to represent the decentralized nature of many distributed systems, multiple agents are required and they will need to interact for the fundamental reason of achieving their individual objectives. Establishing collaborations with other partner agents, they obtain the provision of other services (i.e. social ability). An obvious problem is how to conceptualize systems that are capable of rational behavior. One of the most appreciated solutions to this problem involves viewing agents as *intentional entities*, whose behavior can be predicted and explained in terms of *attitudes* such as belief, desire, and intention (BDI) [20]. In AOP the idea is that, as in declarative programming, we state our goals, and let the built-in control mechanisms figure out what to do in order to achieve them. In BDI agents the computational model corresponds to the human intuitive understanding of beliefs and desires, and so the designer needs no special training to use it.

Another optional, but equally powerful, feature of software agents is *mobility* [10]. Conventional distributed systems assume that the various portions of the distributed application run on their own network node and are bound to it for their whole life. *Mobile Agents (MA)* reshape the logical structure of distributed systems, by providing a system in which components can dynamically change their location, migrating with them a part or the entire agent's state [4], [5].

### B. Service-Oriented Computing

*Service-Oriented Computing* (SOC) [23] proposes a logical view of a software system as a set of *services*, provided to end-users or other services. This recent paradigm proposes itself as the next evolutionary step of the client-server architecture applied to the modern highly distributed and dynamic business scenario.

SOC has the purpose of unifying business processes modularizing large applications into services. Any client, independently of its operating system, architecture or programming language, can access the services in the Service-Oriented Architecture (SOA) and compose them in more sophisticated business processes. *Reusability* has the benefit of lowering development costs and speeding time to market, but achieving high reusability is a hard task. SOC emphasizes the reuse of services, which have to be created *agnostic* to both the business and the automation solutions that utilize them; in addition they need to preserve the maximum degree of *statelessness* towards their current requestor. Moreover, one of the key concepts proposed by service-orientation is *loose-coupling* between the entities playing the role of client and server: limiting service dependencies to the *service contract* allows the underlying provider and requestor logic to remain loosely coupled. *Web Services* technology is, without any doubts, the most promising and industry-supported standard technology, adopted to implement all the service-orientation design principles, discussed more deeply throughout this paper.

## III. Two Approaches to Deal with Complexity

The evolution of distributed software development has been largely driven by the need to accommodate increasing degrees of dynamicity, decentralization and decoupling between distributed components. Software paradigms should take care of the new requirements of distribution and handle complexity from the early stages of the design. In the following subsections, we are going to analyze some aspects, related to the development of complex distributed systems [14], showing what design-level tools the two compared paradigms provide to the designer.

### A. Decomposing the Problem Space

Experience in software engineering taught that complex systems are inherently decomposable and many details can be ignored in the higher-level representations, thus limiting the scope of interest of the designer at a given time. Model entities can be grouped together and their relationships described trying to always provide the highest degree of autonomy between the components.

In the *Service-Oriented Computing* (SOC) paradigm, decomposition is based on the concept of services, which encapsulate units of logic that can be small or large. Service logic can encompass the logic provided by other services, when one or more services are composed into a collection. A typical automation solution is represented by a business process, whose logic is decomposed into a series of steps that execute in predefined sequences according to business rules and runtime conditions. Services can be designed to encapsulate a task performed by an individual step or a sub-process comprised of a set of steps.

*Agent-Oriented Programming* proposes an approach in which the problem space should be decomposed introducing multiple, autonomous components (i.e. agents) that can act and interact in a flexible way to achieve their set of goals. This sort of goal-driven decomposition has been even acknowledged by object-oriented community [16] as being more intuitive and easier than decomposition based on objects. It means that individual agents should localize and encapsulate their own control: in other words, they should be *active*, owning their thread of control, and *autonomous*, taking exclusive control over their own actions.

Although distributing automation logic is nothing new, the two approaches are both stressing the importance of *loose-coupling* when designing distributed components. The first inadequacy of previous paradigms derives from allowing components to form tight connections that result in constrictive inter-dependencies. By decomposing businesses into self-contained and loosely-coupled services, SOC helps achieving the key goal of being able to respond to unforeseen changes in an efficient manner: a service acquires knowledge of another one by means of *service contracts*; interactions take place only with predefined parameters, but the two services still remain independent of each other. In AOP, software agents are likewise self-governing entities, with well-defined boundaries and interfaces, situated in an environment over which they have partial control and observability; their encapsulated state is not accessible to other agents and mutual interactions occur by means of some kind of agent communication language (ACL).

Nevertheless, the strategies adopted by the two approaches make them inherently different. SOC keeps following the well-established *functional* philosophy, while Multi-Agent Systems (MASs) can be classified as *reactive* software systems. A service-oriented business process is started by the action of the end-user or by another client process; it performs its computations following a predefined execution flow (possibly invoking other services inside or outside the enterprise boundaries) and returns the results to the caller. In contrast, MASs are reactive because their components (i.e. agents) often do not terminate, but rather maintain ongoing interactions with their environment. Such interactions are also characterized by *proactiveness*, since an agent tries different ways to achieve its goals and is, consequently, able to influence its environment.

It is commonly agreed that the natural way to modularize most complex systems is in terms of multiple autonomous components, acting and interacting in flexible ways and exhibiting a goal-directed behaviour. This makes perhaps the agent-oriented approach the best fit to this ideal. Moreover, although the intrinsic complexity of the *functional* mapping may be great (e.g. in the case of very dynamic systems, such as air traffic control systems), functional programs are, in general, simpler to specify, design and implement than reactive ones.

### *B. Modelling Interactions*

It was argued that distributed applications are increasingly built out of highly decoupled components. Nevertheless, components need to interact to achieve the required behaviour. Interactions pose some demanding issues at design time, related mainly to the *nature of interactions*, the *degree of flexibility* to provide and the *location* of the interacting entities, which we are going to detail in the following subsections.

*1) Nature of Interactions:* With regard to the *nature of interactions*, the service-oriented paradigm is still more bound to the past than the agent-oriented one. Services interact with each other almost barely at a “syntactic level”, with one service invoking an operation exposed by another one and, after a given time, retrieving the produced result. We said “almost”, because, compared to the classical “method invocation” philosophy provided by OO systems, SOC gives increased importance to the dynamic selection and binding of operations. Using a special intelligent lookup service (e.g. the UDDI registry), a component can search for another service that satisfies a set of key requirements, such as quality of service, accuracy of results or response time. This kind of enhanced reflection technique becomes a fundamental asset from the standpoint of robustness and flexibility: it allows components to dynamically select or reconfigure their bindings, saving a proper amount of independence with respect to the traditional static binding approach. Service discoverability can be considered a promising semantic evolution of the OO polymorphism, since services are expected to match based more on their semantics (e.g. service behaviour under certain conditions, delays, reliability, etc.) rather than on their syntax (e.g. operation prototypes, parameter types, etc.). However, although currently the publisher can define certain service policies to express preferences and assertions about the service behaviour, research efforts are currently underway to continually extend the semantic information provided by service description documents.

The agent-oriented paradigm definitely chooses an interaction model based on semantics and human sociality. Software agents interactions occur at the knowledge level, through a declarative communication language, inspired by the speech act theory [8]. Interacting via this kind of agent-communication language, an agent has the capability to engage in social activities, such as cooperative problem solving or negotiation. The sequence of actions performed by an agent is therefore not statically defined, but depends mainly on its goals and on the environment where it lives. In AOP, the idea is that, as in declarative programming, the designer states her goals and lets the built-in control mechanism figure out what to do, at what time and by whom, in order to achieve them. Moreover, the resources available in the environment can modify the kind of action performed. The control mechanism implements some computational model, like the BDI model, which is undoubtedly more intuitive to the designer than the procedural model.

It must be pointed out that, increasing the abstraction level of interactions, as AOP does, introduces new challenging issues. For agents to interact productively, they must have a bit of knowledge about the expected behaviour of interacting partners, as well as the passive components of their environment. A consistent development effort must focus on modelling the environment, the world in which agents operate and of whom they have beliefs. Knowledge representation languages [17] are proving to be a promising way of describing the environment model, so that social agents can act and interact starting from common views of the world.

Unfortunately, the agents’ models will be often mutually incompatible in syntax and semantics, thus stressing the importance for semantic reconciliation (e.g. by means of some kind of ontology composition technique [26]).

*2) Flexibility of Interactions:* Although many complex systems are decomposable, complexity means also that it is impossible to know a priori about all potential links between the distributed components: interactions often occur at unpredictable times and trying to consider all the possibility at design time is a futile effort.

AOP provides a high degree of *flexibility* as regards the engineering of complex systems: it adopts the policy of deferring to runtime decisions about component interactions, endowing agents with the ability to initiate interactions and deal with unanticipated requests in a flexible manner.

Services, even if carefully designed, cannot provide such level of dynamic interactions, because they are passive entities with respect to agents. In the SOA world, interactions have to be planned and coordinated using choreography or orchestration techniques, recently standardized as Web Service extensions. The WS-BPEL (Web Services Business Process Execution Language) is an example of how a business process can be governed, specifying which services should be called, in which sequence and at what conditions.

*3) Interactions and Component Location:* When thinking about the architecture of a distributed application, interactions among the various components are usually considered independent of the components' location. Location is simply regarded as an implementation detail. Many technologies, such as CORBA, intentionally hide the location of components, making no distinction between interactions of components residing on the same host and components scattered among distant network nodes. However, in many distributed applications, location needs to be considered also during the design stage, since interactions can be remarkably different in terms of latency, partial failure and concurrency.

In the SOC model, services interact with each other, exchanging information through a communications framework that is capable of preserving a loosely-coupled relationship. This framework is based on *messaging*. Messages (e.g. expressed using SOAP protocol) are formatted following the service contract specifications in order to be correctly understood and processed by the target service. The size and number of the exchanged messages depend on the service contract and can be significant when big pieces of information must be exchanged or complex interactions are carried out. Moreover, the widespread use of wireless networks is making available communication channels with low bandwidth or reliability. The design of distributed applications becomes therefore more complex, in that it must aim at avoiding as much as possible the generation of traffic over the weaker links.

The SOC paradigm has only one way to achieve this goal, that is, to increase the granularity of the offered services. In this way, a single interaction between client and server must be sufficient to specify a large number of lower level operations, which are performed locally on the target service and do not need to pass across the physical link. Coarse granularity reduces dependencies between the interacting parts and produces fewer messages of greater significance [12]. Furthermore, the trend to create interfaces for the services that are coarser than those traditionally designed for RPC-based components has been encouraged by vendors as a means of overcoming some of the performance challenges associated with XML-based processing (e.g., SOAP messages are XML-based documents). However, the coarser the granularity of an interface, the less reuse it may be able to offer. If multiple functions are bundled in a single operation, it may be undesirable for clients that only require the use of one of those functions. Then, service interface granularity is a key strategic decision point that deserves a good deal of attention during the design phase.

Mobile Agents (a special kind of software agents presented in Section 2.1) could help because they allow, by their nature, to specify complex computations that can move across a network [10]. Hence, the services that have to be executed by a server that resides in a portion of the network, reachable only through an unreliable and slow link, could be described using a mobile agent; this agent, thanks to its mobility, can be injected into the destination network, thus passing once through this link. There, it could execute autonomously, needing no more connection with the node that sent it, except for the transmission of the final results of its computation (i.e. *disconnected operations*).

### C. Component Reuse and Customization

Nowadays, business process automation is proving that the centralization of control, persistency and authorization is often inefficient, impractical or simply inapplicable. The so called "buy vs. build" or "incremental development" is becoming more and more valid, albeit interpreted in the new scenarios: only a part of the components involved in a distributed computation are under the control of the designer, while the rest may be pre-existing off-the-shelf components that is mandatory or convenient to exploit. If the reuse of components is thus an important aspect, their customization is fundamental as well: an extensible service is likely to be reused more than a rigid one, since it is expected to meet the requirements of much more users/clients. In the following we shall analyse these two aspects.

*1) Service Reusability:* One of the great promises of SOC is that service reuse will lower development costs and speed up time to market. Service-orientation encourages reuse in all services, regardless whether immediate requirements for reuse exist. By applying design standards that make each service potentially reusable, the chances of being able to accommodate future requirements with less development effort are increased. If the service is designed following the SOC principles of autonomy and loose coupling, these weaker dependencies make the applicability of its functionality broader. Furthermore, if the service is designed to be *stateless*, this helps promoting reusability as well as great scalability: if a service is responsible for retaining activity state for long periods of time, its ability to remain available to other requestors will be impeded. Service statelessness supports reuse because it maximized the availability of a service and typically promotes a generic service design that defers activity-specific processing outside service logic boundaries. In turn, the reusability requirement facilitates all forms of reuse, including inter-application interoperability and composition (e.g. service orchestrations and choreographies).

The AOP view promotes instead reusability in different ways. Rather than stopping at reuse of subsystem components and rigidly preordained interactions, agents enable whole subsystems and flexible interactions to be reused within and between applications. Flexible interaction patterns, such as those enabled by the BDI model, and various forms of resource-allocation and auctions patterns have been reused in a significant number of applications.

*2) Service Customization:* As already said, any service operation in a SOA can be invoked and returns results using, for example, SOAP messaging. Message structure has been carefully thought to enable service reusability and customizability: the idea is to equip the message with embedded processing instructions and business rules, which allow them to dictate to recipient services how they should be processed. These allow messages to become increasingly self-reliant by grouping metadata details (in the SOAP header) with message content into a single package (the SOAP envelope). The processing-specific logic embedded in a message alleviates the need for a service to contain this logic. In other words, services in the SOC view should adapt their behavior to the requirements of their current clients, in order to provide the greatest chances of reuse. As a consequence, SOC imposes that service operations become more generic and less activity-specific. The more generic a service's operations are, the more reusable the service.

Likewise, agent mobility encourages the implementation of more generic, and thus highly reusable, service providers [6]. Servers providing an a priori fixed set of services accessible through a statically defined interface are inadequate in those distributed scenarios where new clients can request unforeseen operations at any time. Upgrading the server with new functionalities is only a temporary and inefficient solution, since it increases complexity and reduces flexibility. Mobile code technologies enable a scenario in which the server actually provides a unique service: the execution of mobile code. This feature allows the user to customize and extend the services according to its current needs, bringing the know-how (i.e., the method code) along its way roaming the network.

The possibility of customization granted by mobile code paradigms is therefore more powerful and expressive compared to embedding processing logic in SOAP messages; this, however, comes at the cost of an intrinsic fragility of the execution environment hosting external mobile agents (not only in the case of intentionally malicious agents, but also in the case of bad-designed or misbehaving code [25]).

### *D. Coping with Interoperability and Heterogeneity*

The problem of interoperability between heterogeneous technologies is gaining great interest in the academia but, first and foremost, among the major software vendors.

Web services standards have demonstrated the power of standardization and platform-vendor neutrality. The emerging SOC paradigm took the principle of openness of standards as one of its foundation stones: the cost and effort of cross-application integration are significantly lowered when applications being integrated are SOC-compliant.

The landscape of agent-oriented software seems to be more fragmented, although many efforts towards the definition of standards are growing in the research community. Several (Mobile) agent platforms have been developed, but one of the weaknesses of those platforms is the lack of true interoperability, being the strength of Service-Oriented systems. Many of the proposed agents platforms provide support for migration, naming, location and communication services, but they differ widely in architecture and implementation, thereby impeding interoperability and rapid deployment of mobile agent technology in the marketplace. To promote interoperability, some aspects of mobile agent technology have been standardized. Currently there are two standards for mobile agent technology: the OMG's Mobile Agent System Interoperability Facility (MASIF) and the specifications promulgated by the Foundation for Intelligent Physical Agents (FIPA). MASIF [15] is based on agent platforms and it enables agents to migrate from one platform to another, while FIPA [9] is based on remote communication services. The former is primarily based on mobile agents travelling among agent systems via CORBA interfaces and does not address inter-agent communications. The latter focuses on intelligent agent communications via content languages and deals with the mobility aspect of agents only since the FIPA 2000 release. In order to achieve the degree of outstanding platform neutrality and interoperability of SOC, some research programs are studying the possibility of an integration of MASIF/FIPA specifications into a commonly agreed standard for MAPs [1].

### *E. Security*

When application logic is spread across multiple physical boundaries, implementing fundamental security measures such as authentication and authorization becomes more difficult. In the traditional client-server model, the server is the owner of any security information, needed to recognize user's credentials and to assign privileges for the use of any protected resources. Well-established techniques, such as SSL (Secure Socket Layer), granted a so-called *transport-level* security, where the whole channel, by which requests and responses are transmitted, is protected.

SOC departs from this model by introducing substantial changes to how security is incorporated and applied. Relying heavily on the extensions and concepts established by the WS-Security framework, the security models used within SOC emphasize the placement of security logic onto the *messaging level*. SOAP messages provide header blocks in which security logic can be stored (e.g. by means of X509 certificates). So, wherever the message goes, so its security information does. This approach is required to preserve individual autonomy and loose coupling between services, as well as the extent to which a service can remain fully stateless.

The "mobility" concept in the agent-oriented paradigm poses new and more challenging security issues. Moving code, in addition to data, brings security problems that fit into two main categories: protecting host systems and networks from malicious agents and protecting agents form malicious hosts. Digital signatures and trust-management approaches may help to identify the agents and how much they should be trusted. The malicious host that attacks a visiting mobile agent is the most difficult and largely unsolved: such a host can steal private information from the agent or modify it to misbehave when it jumps to other sites.

Security is perhaps the most critical factor that has limited a widespread acceptance of the mobile agent paradigm for strategic applications, such as e-commerce, where sensitive transactions have to be performed with the highest level of security. Services interact with each other without moving any piece of application logic (e.g. a thread), but simply moving inert parameters data to invoke exposed service operations. Protecting passive data is usually more straightforward than protecting mobile code, albeit several research efforts [24] are being made to reduce the gap between the two approaches.

## IV. Integrating Services and Agents

We have analyzed, throughout this paper, some of the similarities and differences, strengths and weaknesses of two emerging paradigms in distributed software engineering: Service-Oriented Computing and Agent-Oriented Programming. It has been observed that they tackle complexity in software, often from very distant points of view, promising advantages to designers but also introducing architectural headaches. We are convinced that the silver bullet of distributed software paradigms cannot be identified in any of these individual paradigms: distributed systems in the future will likely benefit of ideas drawn from both of them, but this demands for some intelligent form of integration of the two approaches.

In the recent years, the emphasis of service-oriented architectures has been on the execution of services, as building blocks to decompose and automate complex and distributed business problems. The Web Service infrastructure is widely accepted, standardized, and is likely to be the dominant technology over the coming years. However, the next evolutionary step for services will be driven by the need to deal with target environments that become even more populous, distributed and dynamic. Therefore, many approaches are emerging for the future of these models and they all agree on one point: the integration between services and agents is more than feasible. For example, in the last issue of the AgentLink III Agent Technology Roadmap, Web Services are presented “as “a ready-made infrastructure that is almost ideal for use in supporting agent interactions in a multi-agent system” [2]. In this direction, different approaches to integration have been proposed and, in many cases, tested with some prototypal applications.

A first idea of integration consists in the mere enabling of interactions between the two worlds. In other words, some researchers [11] have experimented techniques to make agents and web services interoperate. In order to make web services invoke agent capability and vice versa, these systems try to formalize a proper mapping between the WSDL service contract and the Agent Communication Language (ACL). These approaches, however, have been criticized, since they try to blur the distinction between agency and service-oriented concepts: if agents are accessed through pre-defined, fixed interface operations (accepting parameters and returning results), they are implicitly treated as services, and as a consequence they loose the autonomy and intelligence belonging to agents. Vice versa, if a service behaves in a non-deterministic way and other services must interact with it using some high-level ACL, this service should be regarded conceptually as an agent instead. A more clear integration approach [7], [19] recognizes the conceptual difference between agents and services and proposes a functional layered view of their interactions. Services are the functional building blocks, which can be composed to form more complex services, but they remain passive entities used by agents in distributed applications: the agent are given some high-level goals; they are primarily responsible for adopting strategies or plans and translate them into concrete actions, such as invoking an atomic service or composing other services into new functional aggregates. In a few words, many researchers are expressing the relationship between services and agents, saying that services provide the computational resources, while agents provide the coordination framework [3].

A new research roadmap [13] is proposing a radical evolution of the concept of service, rather than an integration: the main idea is to give more “life” to services so that, instead of passively waiting for discovery, they could proactively participate in the distributed application, just like agents in multi-agent systems. Making services increasingly alive and enabling more dynamical interactions, services are expected to function as computational mechanism, enhancing our ability to model and manage complex software systems. As already said, a service knows only about itself, but not about its clients; agents are self-aware, but gain awareness of the capabilities of other agents as interactions among agents occur. Equipped with such an awareness, it has been advised that a service would be able to take advantage of new capabilities in its environment and could customize its service to a client, for example, improving itself accordingly.

## V. Conclusions

This paper presented a comparison between the Services-Oriented vision and the (Mobile) Agents one, as concerns the several issues in the development of complex distributed systems. We pointed out that each approach offers its own advantages, but some rules of thumbs emerge from the comparison: model entities of SOC are better fitting “closed” distributed systems, where the components are explicitly designed to organize themselves in a predefined (i.e. choreographic or orchestrated) fashion to achieve the fulfillment of a certain business process or workflow. “Open” systems are instead better manageable if a mobile agent approach is taken: in application scenarios, like pervasive computing or online auctions, it can be impossible to know a priori all potential interdependencies between components (what services are required at a given point of the execution and with what other components to interact), as a functional-oriented behavior perspective typically requires. In the latter case, agents can consider also the possibility of competitive behavior in the course of the interactions and the dynamic arrival of unknown agents.

Nevertheless, we think that an integration of the two paradigms is more than desirable. In our vision, services constitute an established, platform-neutral and robust computational infrastructure, made up of highly reusable building blocks, from which a new breed of distributed software paradigms, derived from the agent-oriented world, can emerge. This new phase seems to be already started, for example if we look at the research in the field of autonomic services [18], where researchers are exploring the possibility of embedding some form of self-management in the components that will provide the services of the future.

## ACKNOWLEDGMENT

Work supported by the Italian MiUR in the frame of the PRIN project “MEnSA—Agent oriented methodologies: engineering of interactions and relationship with the infrastructures”.

## REFERENCES

[1] http://olympus.algo.com.gr/acts/dolphin/AC-baseline.html

[2] http://www.agentlink.org/roadmap/index.html

[3] P. A. Buhler and J. M. Vidal, *Towards adaptive workflow enactment using multi-agent systems*, in Information Technology and Management, 6:6187, 2005.

[4] G. Cabri, L. Leonardi, M. Mamei, F. Zambonelli, *Location-dependent Services for Mobile Users*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems And Humans, Vol. 33, No. 6, pp. 667–681, November 2003.

[5] G. Cabri, L. Ferrari, L. Leonardi, R. Quitadamo, *Strong Agent Mobility for Aglets based on the IBM JikesRVM*, in the Proc. of the 21st Annual ACM Symposium on Applied Computing (SAC), Dijon, France, April 23–27, 2006.

[6] G. Cugola, C. Grezzi, G.P. Picco and G. Vigna, *Analyzing Mobile Code Languages*, Mobile Object Systems n. 1222, Springer, 1997.

[7] I. Dickinson, M. Wooldridge, *Agents are not (just) web services: considering BDI agents and web services*, in Service-Oriented Computing And Agent-Based Engineering (SOCABE ’05), 2005.

[8] T. Finin, Y. Labrou, J. Mayfield, *KQML as an agent communication language*, in J. Bradshaw, ed., Sofware agents, MIT Press, Cambridge, MA, 1995.

[9] http://www.fipa.org/

[10] A. Fuggetta, G. P. Picco, G. Vigna, *Understanding Code Mobility*, IEEE Transactions on Software Engineering, Vol. 24, 1998.

[11] D. Greenwood, M. Calisti, *An Automatic, Bi-Directional Service Integration Gateway*, In Proc. of the Workshop on Web Services and Agent-Based Engineering (WSABE 2004), 2004.

[12] M. Huhns, M. P. Singh, *Service-Oriented Computing: Key Concepts and Principles*, in IEEE Internet Computing, Service-Oriented Computing Track, January-February 2005.

[13] M. Huhns, M. P. Singh et al., *Research Directions for Service-Oriented Multiagent Systems*, in IEEE Internet Computing, November–December 2005.

[14] J. Kramer, *Distributed Software Engineering*, In Proc. of the 16th International Conference on Software Engineering, Sorrento (Italy), May 1994.

[15] Mobile Agent System Interoperability Facility specifications (MASIF), http://www.omg.org/docs/orbos/97-10-05.pdf

[16] B. Meyer, *Object-Oriented Software Construction*, Prentice Hall, 1988.

[17] A. Newell, *The knowledge level*, in Artificial Intelligence 18, 1982.

[18] H. Liu, V. Bhat, M. Parashar and S. Klasky, *An Autonomic Service Architecture for Self-Managing Grid Applications*, in the Proc. of the 6th IEEE/ACM International Workshop on Grid Computing (Grid 2005), Seattle, USA, November 2005.

[19] M. Pistore, F. Barbon, P. Bertoli, D. Shaparau and P. Traverso, *Planning and Monitoring Web Service Composition*, In Workshop on Planning and Scheduling for Web and Grid Services, 2004.

[20] A. Rao and M. Georgeff, *BDI Agents: From Theory to Practise*, In the Proc. of the 1st International Conference on Multi-Agent Systems (ICMAS-95), 1995.

[21] L. Ruimin, F. Chen and H. Yang, *Agent-based Web Services Evolution for Pervasive Computing*, in the Proc. of the 11th Asia-Pacific Software Engineering Conference (APSEC’04), 2004.

[22] M. Shaw and D. Garlan, *Software Architecture: Perspective on an Emerging Discipline*, Prentice Hall, 1996.

[23] M. P. Singh and M.N. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*, John Wiley and Sons, 2005.

[24] C. F. Tschudin, *Mobile agent security*, In Intelligent Information Agents, Springer-Verlag, 1999.

[25] G. Vigna, *Mobile Agents: Ten Reasons For Failure*, in the Proc. of the 2004 IEEE International Conference on Mobile Data Management (MDM’04), Berkeley, California, USA, January 2004

[26] G. Wiederhold, *An Algebra for Ontology Composition*, in Proc. of Monterey Workshop on Formal Methods, pp. 56–61, 1994.

[27] M. Wooldridge, *Agent-based software engineering*, in IEEE Proc. of Software Engineering pp. 26-3-7, 1997.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Connecting Methodologies and Infrastructures in the Development of Agent Systems

Giacomo Cabri, *Member, IEEE*, Mariachiara Puviani, *Member, IEEE* and Raffaele Quitadamo
Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia
Via Vignolese, 905 41100 Modena – Italy
Email: {giacomo.cabri, mariachiara.puviani, raffaele.quitadamo}@unimore.it

***Abstract*—In the building of agent systems developers can be supported by both appropriate methodologies and infrastructures, which guide them in the different phases of the development and provide useful abstractions.**

**Nevertheless, we assist to a situation in which methodologies and infrastructures are not connected each other: the products of the analysis and design phases could not always be exploited in the implementation phase in a direct way, even if sometimes CASE-tools are present to help in translating methodologies' diagrams in infrastructures' code. This leads to a "gap" between methodologies and infrastructures that is likely to produce fragmented solutions and to make the application maintenance difficult.**

**In this paper we face this issue, proposing three directions to solve the problem. We do not want to propose a "new brand" methodology and infrastructure tightly connected, rather, we aim at reusing as much as possible what already exists, not only in abstract terms, but also in concrete "fragments" of methodologies; an appropriate meta-language that describes how a methodologies works would be useful to more easily map them onto the infrastructures, or even to "compose" a new methodologies. A further approach is based on an "intermediate" layer between methodologies and infrastructures, which provides a mapping between the involved entities.**

## I. Introduction

THE demand of effective paradigms for developing complex systems is significantly increasing in the last years [16]. Developers are required to model and manage complex scenarios, often physically distributed; in addition, such a management should be more and more autonomous, in order to take the correct decisions with less human intervention as possible. The development of such intelligent complex systems must be addressed appropriately, with effective models and tools.

Unfortunately, the current situation presents two different approaches that are likely to be in contrast: a *top-down* approach, which follows the traditional software engineering directions by providing *methodologies* that however often discard the implementation phase; a *bottom-up* approach, which meets concrete requirements by providing *infrastructures* that are likely to lack model foundations. This situation leads to a "gap" between methodologies and infrastructures, with the consequence of fragmented solutions in the development of agent systems.

The *software agent* paradigm is one of the most exploited to build complex systems [16]. On the one hand, agents exhibit the features of *proactivity* and *reactivity*, which lead to a high degree of autonomy and a certain degree of intelligence. On the other hand, their *sociality* feature enables the building of systems where agents are distributed and interact to carry out common tasks, as happens in Multi-Agent Systems (MAS).

In our work, we have analyzed several agent methodologies and agent infrastructures, first to confirm the presence of such a gap, then to study some solutions to fill this gap. The aim of this paper is to point out the existing gap and to propose some approaches to fill it. Due to the page limitation and the fact that our work is at its beginning, in this paper we sketch three solutions without giving too many details, providing readers with the "flavor" of the possible directions.

In rest of the paper we start presenting the evaluated methodologies and infrastructures, and discussing about the gap between them (Section II). Then, the core of the paper is represented by the proposal of three possible approaches as solutions of such a gap (Section III). Finally, Section IV concludes the paper.

## II. The Gap between Methodologies and Infrastructures

In this section we briefly introduce the evaluated methodologies and infrastructures, and then point out the existing gap we are going to fill, sketching also some previous work. We remark that the aim of this paper is not to report comparisons: interested readers can refer to our previous work [7], [10], [24].

### A. Methodologies

Among several methodologies for developing agent systems, we have chosen the most spread ones:

- ADELFE (Toolkit for Designing Software with Emergent Functionalities) [4] defines a methodology to develop applications in which self-organization makes the solution emerge from the interaction of its parts, and it guarantees that the software is developed according to the AMAS (Adaptive Multi-Agent System) theory [11]. It is dedicated to the design of systems that are complex, open and not well-specified. It is based on well-known tools and notations coming from the object-oriented software engineering: UML and RUP (Rational Unified Process);

- Gaia [26] was the first methodology proposed to guide the process of developing a MAS from analysis to design, starting from Requirements Statements. It guides developers to a well-defined design for a MAS. A MAS is seen as an organisation of individuals, each of which playing specific roles in that organisation, and interacting according to its role. In its first version it suffers from several limitations: it is suitable only for designing closed MAS, and the notions it uses are not suitable for dealing with the complexity of real-word. Trying to overcome these problems its official extension is Gaia v.2;
- PASSI (Process for Agent Societies Specification and Implementation) is a step-by-step requirement-to-code methodology [12], for designing and developing Multi-agent societies, using the UML standard notation. It aims at using standards whenever it is possible and considers two different aspects of agents: (i) autonomous entities capable of pursuing an objective through autonomous decisions, action and social relationship and (ii) parts of a system. The modelling of requirements is based on use-case, and ontology has a central role in the social model. It aims at reusing existing portions of design code, using a pattern-based approach. Its design process is incremental and iterative;
- Prometheus [22] is a methodology for developing agent-oriented systems, and it aims at covering all the stages of software development [13]. It includes provision of "start-to-end" support; a hierarchical structuring mechanism that allows design to be performed at multiple levels of abstraction; and support for detailed design of the internals of intelligent agents. Prometheus also uses an iterative process over all its phases;
- SODA (Societies in Open and Distributed Agent spaces) [1], [20] is an agent-oriented methodology for the analysis and design of agent-based systems, focusing on inter-agent issues, like the engineering of agent societies and the environment for MAS. Its new version takes into account both the Agents and Artifacts (A&A) meta-model, and a mechanism to manage the complexity of system description;
- Tropos methodology is intended to support all phases of software development [5]; in Tropos, organizational architectural styles for cooperative, dynamic and distributed applications are defined to guide the design of the system architecture. It adopts Eric Yu's i* model which offers the notions of actor, goal, role and actor dependency as primitive concepts for modelling the applications. Tropos has two key features: the notions of agent, goal, plan and various other knowledge level concepts are fundamental primitives used uniformly throughout the software development process; a crucial role is assigned to requirements analysis and specification when the system-to-be is analysed with respect to its intended environment.

### B. Infrastructures

With regard to the agent infrastructures, we have considered the following ones:

- CArtAgO (Common Artifact for Agent Open environment) [25], exploits the concept of *workplace*, an organisational layer on top of *workspaces*. A *workspace* is a set of *artifacts* and *agents*, as a *workplace* is the set of *roles* and *organisational rules* being in force in a workspace. *Roles* can be played by *agents* inside the workplace, and they may or may not give permissions to *agents* to use some *artifacts* or to execute some specific operations on selected *artifacts*.
- JACK [2] is a multi-agent platform based on the BDI model; it is written in Java and is commercial-oriented. Following the BDI model, JACK enables the definition of agents' plans that start from the agents' knowledge and lead to the achievement of the agents' goals.
- JADE [3] (Java Agent DEvelopment framework) implements a FIPA-compliant general-purpose multi-agent platform fully implemented in Java language, that provides an agent-oriented infrastructure. The main concept of Jade is the *agent*, which is implemented by the *AgentJ* class and is associated to an Agent State, a Scheduler, and a Message Queue. Another important concept id the *behaviour*, which defines the main features of an agent. Jade offers a set of graphical tools that supports the debugging and deployment phases;
- MARS (Mobile Agent Reactive Spaces) [9] is a coordination medium based on programmable Linda-like tuple spaces for Java-based mobile agents. MARS allows agents to read and write information in the form of tuples adopting a pattern-matching mechanism. Moreover, its behaviour can be programmed to suit environment or application requirements. It it has been designed to complement the functionality of already available agent systems, and it is not bound to any specific implementation;
- RoleX (Role eXtension) [6] implements an infrastructure to manage roles for agents. It has been implemented in the context of BRAIN (Behavioural Roles for Agent INteractions) framework which proposes an approach where the interactions among agents are based on the concept of role. A *role* is defined as a set of *actions* that an *agent* playing that *role* can perform to achieve its task, and a set of *events* that an *agent* is expected to manage in order to act as requested by the *role* itself;
- TOTA (Tuples On The Air) [17] is a coordination middleware based on tuple spaces for multi-agent coordination, in distributed computing scenarios. TOTA assumes the presence of a network of possibly mobile *nodes*, each running a *tuple space*: each agent is supported by a local middleware and has only a local (one-hop) perception of its environment. Nodes are connected only by short-range network links, each holding references to a (limited) set of neighbour nodes: so, the topology of the network,

as determined by the neighbourhood relations, may be highly dynamic;

- TuCSoN (Tuple Centers Spread Over Networks) [21] is a coordination medium based on Linda-like tuple spaces; it is focused on the the communication and coordination of distributed/concurrent independent agents. In TuCSoN,the *Agent Coordination Context* (ACC) works as a model for the *agent* environment, by describing the environment where an *agent* can interact. It also enables and rules the *interactions* between *agents* and the environment, by defining the space of admissible agent *interactions*. ACC has first to be negotiated by each *agent* with the MAS infrastructure, and then the *agent* specifies which *roles* to activate: if the *agent* request is compatible with the current organisation rules, a new ACC is created, configured according to the characteristics of the specifies *roles*, and is released to the *agent* for active playing inside the organisation.

### C. The Gap

In our work we have studied the chance of integration between methodologies and infrastructures, evaluating whether and how the concepts considered by the methodologies can match with the concepts dealt with in the infrastructures. To this purpose, we exploited meta-models of both methodologies and infrastructures; in fact, the meta-model described the entities that represent the considered concepts of methodologies/infrastructures without looking at the different phases where they are involved, and allows to evaluate whether a matching exists and of which extent.

What emerges from our study is that there is not continuity between the methodologies and the infrastructures, presenting a gap between analysis and design on the one hand and development and implementation on the other hand. This gap concerns in particular the entities that represent the involved concepts. In fact, only few entities can be found in both methodologies and infrastructures with the same meaning, while others are present only in methodologies or infrastructures.

As an meaningful example better detailed in [8], consider the development of an application that simulates an auction. From a first analysis, the developer can design three roles, *bidder*, *auctioneer* and *seller* that can be implemented using an agent infrastructure. From a subsequent analysis, the design of a *broker* role emerges as needed in order to manage the real transactions from bidders and sellers. If there is a separation between the methodology and the infrastructure, i.e., there is not a connection between the "role" entity of the methodology and one or more entities of the infrastructure, the addition of a role could require the re-implementation of the application. Instead, if a connection is present, the modifications flow from the analysis/design to the implementation in a smooth way.

The lack of connections could be natural, since methodologies and infrastructures are likely to start from different needs; but it could lead to some problems; first, resulting in a fragmented development, but, more important, making maintenance very difficult: if a methodology entity has not a corresponding infrastructure entity, its change requires to find out how it was implemented.

As mentioned in the introduction, the reasons of such a gap are likely to derive from the different origins of methodologies and infrastructures: from the one hand, the traditional software engineering approaches follow a *top-down* direction; on the other hand, concrete requirements have called for implementation-oriented solutions providing platforms and frameworks to build applications. This gap could also reflect the twofold origin of the agent paradigm: artificial intelligence from the one hand, and distributed systems from the other hand.

Of course, there could be no need for exploiting infrastructure(s) not connected to a chosen methodology: there are different methodologies that have a natively compliant infrastructure. But we remark that the development of complex systems is likely to require the exploitation of different infrastructures, not always connected with the chosen methodology.

So, our aim is bridging agent methodologies and agent infrastructures in order to achieve a continuous support in the development of systems.

Previously, we have performed some work on specific issues.

As a first attempt, we try to map the methodologies' entities with the infrastructures' ones. To this purpose, first of all we have evaluated the entities common to the different infrastructures. This is useful to understand which the "core" entities are, which will deserve a support by the methodologies. This work is similar to build a common ontology that will semantically maps terms across methodologies and infrastructure, but the result will not cover all the necessary entities. So, we did not rely only on the *names* of the entities, but we spent an effort to map also entities with different names but the same meaning.

For instance, we have considered the PASSI methodology and the RoleX infrastructures, and we have produced a mapping based on their meta-models [8]. Molesini et al. have performed a similar attempt [18], considering one methodology (SODA) and three infrastructures (CArtAgO, TuCSoN and TOTA). As another example from our work, we have focused on the *role* entity, and we have evaluated how it is dealt with in methodologies and in infrastructures, in order to map the different approaches [23].

From these experiences, we have learned two main lessons. The former is that the exploitation of meta-models is very useful to understand the involved entities and their relationships; the latter is that a more global approach is needed.

## III. Proposed Approaches

Starting from the previous considerations, in this section we sketch three possible approaches that can fill the gap. Our work has just begun, so we will provide readers with its first results, which require further study in the future.

There are a lot of different methodologies and infrastructures, so we consider impossible to map each methodology and each infrastructure one in the other; instead we aim at more general solutions to help developer, especially those who have to integrate software agents in an existing application, but also those addressing new systems.

We also discarded the idea of creating a “new brand methodology” and a “new brand infrastructure” that can be used together, because they will be “yet another methodology” and “yet another infrastructure” to add in the context of the existing ones.

### A. *An Intermediate Layer*

The first solution is proposed taking into account existing or legacy components or applications that must be integrated.

This approach consists of defining an intermediate layer, which provides entities that are connected to the methodologies on the one hand and to the infrastructures on the other hand (see Fig. 1).

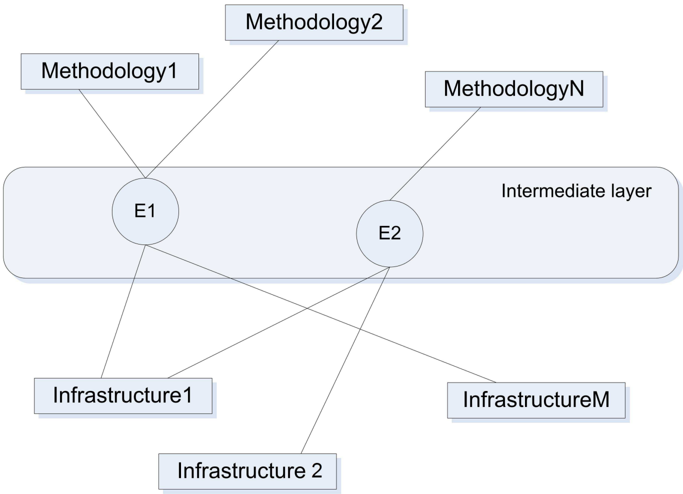


Fig. 1. An intermediate layer

From a particular point of view, this layer can be seen as an ontology, but we prefer to consider it more practically as a “common entities” layer, since it provides all the fundamental entities of the infrastructures chosen for the mapping; we have started from the most important ones, but then we are going to integrate all the others as well.

The entities that emerge from our study not only as common, but also as part of the foundation of the infrastructures, are:

- Agent
- Role
- Action
- Event

This layer can be exploited to map a methodology on every specific infrastructure. First of all, it can map entities starting from a direct mapping (one-to-one) and successively mapping the “composed” entities (one-to-n or n-to-one); during this work the studies of the mapping between methodologies’ entities and infrastructures’ ones previously presented can be very useful.

This new layer will permit to choose the preferred methodology and start to build the applications; then, when the infrastructure has to be chosen, the layer can be used to map the entities of the chosen methodology and the chosen infrastructures.

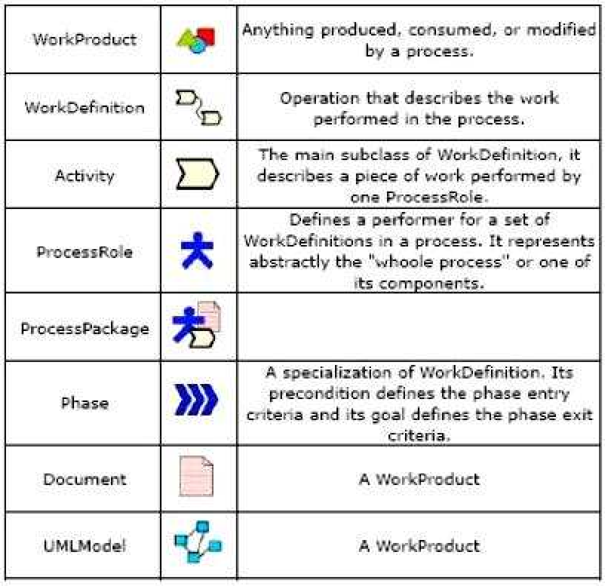
During this phase of translation, an interaction with the developer can be useful because not all the entities can be easily mapped, and sometimes, due to the chosen methodologies or infrastructures, the specifications of an entity by the developer can help the mapping. For example, MARS does not have the “event” concept, but has an “access event”, so the developer can be asked if the event is an access event or another one.

Of course this approach has some limitations, mainly because it is difficult to consider all the existing methodologies and infrastructures in a complete way; but it can be exploited for not forcing developers to choose a particular tool: it permits developers to be free in their choice, and to integrate existing applications.

### B. *SPEM*

An interesting idea to fill the gap is to use the most important features of methodologies and infrastructures to map them together; but it is very difficult because of their different languages and their different phases. If we have the same language for all the methodologies (and infrastructures) it can be possible to make a mapping and even to “select” useful parts of the considered methodologies.

This study can be possible using SPEM (Software Process Engineering Metamodel. In Fig. 2 we briefly present the 1.1 SPEM notation, which has been used for our studies; for further information see also [15] and [19].



| | | |
|---|---|---|
| WorkProduct | | Anything produced, consumed, or modified by a process. |
| WorkDefinition | | Operation that describes the work performed in the process. |
| Activity | | The main subclass of WorkDefinition, it describes a piece of work performed by one ProcessRole. |
| ProcessRole | | Defines a performer for a set of WorkDefinitions in a process. It represents abstractly the "whoole process" or one of its components. |
| ProcessPackage | | |
| Phase | | A specialization of WorkDefinition. Its precondition defines the phase entry criteria and its goal defines the phase exit criteria. |
| Document | | A WorkProduct |
| UMLModel | | A WorkProduct |

Fig. 2. The SPEM notation (v. 1.1)

With this “language” it is possible to describe processes and their components in term of pieces of process called “fragments”, using meta-models. This approach has been used by the OMG specifications, in the context of "FIPA Methodology Technical Committee" project [19], to translate some well known methodologies, like Adelfe, Gaia, Tropos, Passi, etc. in a common language; and the work is still going on with other methodologies. This approach can be used also for infrastructures if there is a clear meta-model to be used for the translation, even if today it is used only for methodologies.

SPEM specification is structured as a UML profile, and provides a complete MOF (Meta Object Facility)-based metamodel. Each fragment that is created is composed of (not necessary all of them) [15]:

1) A portion of process defined with a SPEM diagram;
2) One or more deliverables (artifacts like AUML/UML diagrams, text documents, and so on);
3) Some preconditions which said that it is not possible to start the process specified in the fragment without the required input data or without verifying the required guard condition;
4) A list of concepts (related to the MAS meta-model) to be defined (designed) or refined during the specified process fragment;
5) Guideline that illustrates how to apply the fragment and best practices related to that;
6) A glossary of terms used in the fragment;
7) Composition guidelines—A description of the context/problem that is behind the methodology from which the specific fragment is extracted;
8) Aspects of fragment: textual description of specific issues;
9) Dependency relationships useful to assemble fragments.

The main idea of SPEM is that a software development process is a collaboration between abstract active entities (*process roles*) that perform operations (*activities*) on concrete entities (*work products*).

SPEM can be used to translate all the different methodologies and infrastructures, to help developers to choose which to use for their applications.

Today, the most used version of SPEM is 1.1, but the 2.0 version is just be implemented, and there is also a tool that can validate the created fragments and their integration.

### C. A Composed Methodology

The last solution starts from the SPEM one and is the more radical, but it could be the most effective in the long term.

Given that the difficulties of the translation of all the methodologies and infrastructures is a very long work, a possible approach is to create a "composed methodology" that can be useful for developers, starting from SPEM fragments. This direction has been proposed also by the FIPA Methodology Technical Committee [14]. They have provided a repository of fragments to choose to compose an ad hoc methodology, adding the possibility to create new fragments.

To this purpose, we have selected some important fragments from the existing ones. In our study we have focused on (i) the common processes and entities of the methodologies and (ii) the entities that enable a connection with the main entities of the infrastructures. The composed methodology relies on different important fragments of ADELFE, PASSI, Gaia, Premetheus.

We have defined three phases: *Requirement* (Fig. 3), *Analysis* (Fig. 4) and *Design* (Fig. 5), while the *Implementation* phase is on going work.

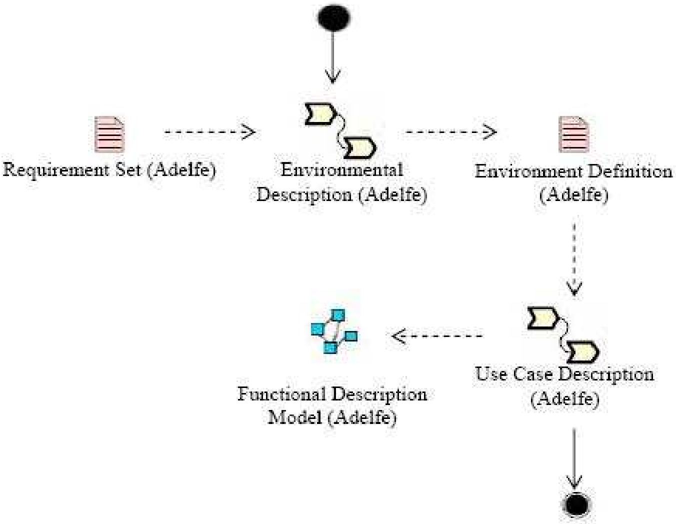

Requirement Set (Adelfe)
Environmental Description (Adelfe)
Environment Definition (Adelfe)
Functional Description Model (Adelfe)
Use Case Description (Adelfe)

Fig. 3. Requirement phase

Some of these different fragments are well integrated, even if they come from different methodologies, while other fragments have been changed inside to match input and output of the different phases. This work has some difficulties because, as just said, entities with the same name but coming from different methodologies can not be used in the same way (e.g. the concept of ”agent” in ADELFE is adaptive, in PASSI is FIPA-like and in Prometheus is a BDI one). The implementation and validation of this composed methodology is still going on, trying to adjust the different fragments.

There are some advantages in exploiting fragments of existing methodologies. First, it is not "yet another methodology", because it takes concrete parts of the other methodologies, not only ideas and concepts; then, the exploited fragments have been tested by developer for different years and in different scenarios; further, there is already related documentation, case studies, tools and practice; finally, some developers can find some parts of the methodology they are used to.

Even if this third solution is more radical, it does not completely discard the existing background, on the contrary it tries to reuse (parts of) existing solutions and experiences.

## IV. Conclusions

In this paper we have addressed the gap existing between methodologies and infrastructures when developing agent systems.

To fill this gap, we have proposed three possible directions. The first relies on an *intermediate layer*, which provides some common entities that are mapped to both methodologies' and infrastructures' ones. The second exploits SPEM, to

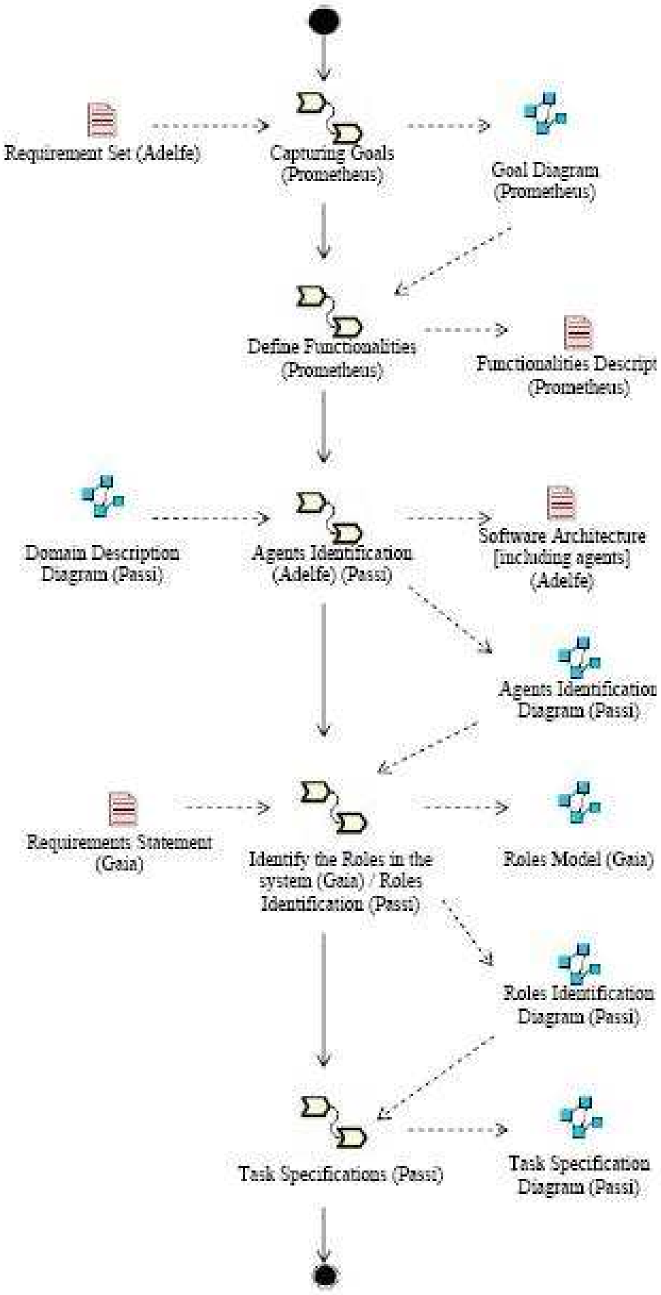

Requirement Set (Adelfe)
Capturing Goals (Prometheus)
Goal Diagram (Prometheus)
Define Functionalities (Prometheus)
Functionalities Descript (Prometheus)
Domain Description Diagram (Passi)
Agents Identification (Adelfe) (Passi)
Software Architecture [including agents] (Adelfe)
Agents Identificatior Diagram (Passi)
Requirements Statement (Gaia)
Identify the Roles in the system (Gaia) / Roles Identification (Passi)
Roles Model (Gaia)
Roles Identification Diagram (Passi)
Task Specifications (Passi)
Task Specification Diagram (Passi)

Fig. 4. Analysis phase

describe both methodologies and infrastructures by a common language, in order to make the mapping easier. Finally, the third approach proposes a new methodology, which is not “yet another methodology”, instead it is a “composed methodology” of fragments extracted from existing methodologies. The proposed solutions range from the most conservative one (useful when existing components are to be integrated with a reduced effort) to the most innovative one (more useful in the development of new systems), even if we have spent an effort to not discard the existing knowledge in the field.

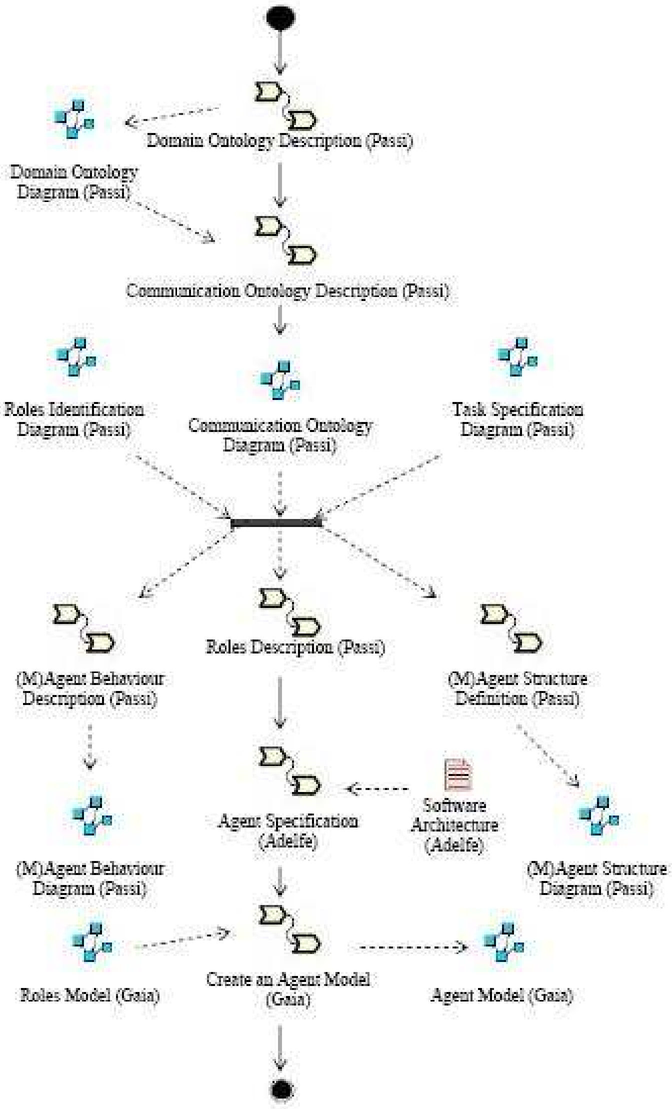

Domain Ontology Description (Passi)
Domain Ontology Diagram (Passi)
Communication Ontology Description (Passi)
Roles Identification Diagram (Passi)
Communication Ontology Diagram (Passi)
Task Specification Diagram (Passi)
(M)Agent Behaviour Description (Passi)
Roles Description (Passi)
(M)Agent Structure Definition (Passi)
(M)Agent Behaviour Diagram (Passi)
Agent Specification (Adelfe)
Software Architecture (Adelfe)
(M)Agent Structure Diagram (Passi)
Roles Model (Gaia)
Create an Agent Model (Gaia)
Agent Model (Gaia)

Fig. 5. Design phase

The approaches sketched in this paper deserve still a lot of future work.

With regard to the *intermediate layer*, we must integrate also other entities beside the ones mentioned in Section III-A, and we must propose and test the possible mappings.

SPEM has recently been updated to version 2.0, promising different improvements that must be evaluated, considering also all the legacy work on this meta-language.

Also the definition of a *composed methodology* can benefit from the new version of SPEM, in particular form the capability of checking the correctness and the completeness of fragment composition.

Finally, whichever the chosen direction, it must be appropriately tested, possibly with formal tools, but certainly with concrete case studies.

## Acknowledgment

Work supported by the Italian MiUR in the frame of the PRIN project "MEnSA—Agent oriented methodologies: engineering of interactions and relationship with the infrastructures".

## References

[1] aliCE Research Group. SODA home page. http://soda.alice.unibo.it

[2] AOS Autonomous Decition-Making Software. Jack agent platform. http://www.agent-software.com/ 2008.

[3] F. Bellifemine. Developing multi-agent systems with jade. In *Proceedings of PAAM 99, London (UK)*, pages 97–108, 1999.

[4] C. Bernon, M.P. Gleizes, G. Picard, and P. Glize. The Adelfe Methodology For an Intranet System Design. In *Proc. of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS), Toronto, Canada*, 2002.

[5] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. A knowledge level software engineering methodology for agent oriented programming. In *Proceedings of the fifth international conference on Autonomous agents*, pages 648–655. ACM Press New York, NY, USA, 2001.

[6] G. Cabri, L. Ferrari, and L. Leonardi. Enabling mobile agents to dynamically assume roles. In *Proceedings of the ACM Symposium on Applied Computing, Melbourne (USA), March*, pages 56–60, 2003.

[7] G. Cabri, L. Leonardi, and M. Puviani. Service-Oriented Agent Methodologies. In *Proceedings of the Sixteenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007*, 2007.

[8] G. Cabri, L. Leonardi, and M. Puviani. Methodologies and Infrastructures for Agent Society Simulation: Mapping PASSI and RoleX. In *Proceedings of the $2^{nd}$ International Symposium "Agent Based Modeling and Simulation", at the $19^{th}$ European Meeting on Cybernetics and Systems Research (EMCSR 2008), Wien, March 2008*, 2008.

[9] G. Cabri, L. Leonardi, and F. Zambonelli. MARS: a programmable coordination architecture for mobile agents. *Internet Computing, IEEE*, 4(4):26–35, 2000.

[10] Giacomo Cabri, Letizia Leonardi, and Mariachiara Puviani. Methodologies for Designing Agent Societies. In *The Second Workshop on Engineering Complex Distributed Systems (ECDS 2008), Barcelona Spain, March 2008*, 03 2008.

[11] D. Capera, J.P. George, M.P. Gleizes, and P. Glize. The AMAS theory for complex problem solving based on self-organizing cooperative agents. In *Proceedings of the Twelfth IEEE International Workshops onE nabling Technologies: Infrastructure for Collaborative Enterprises, 2003*, pages 383–388, 2003.

[12] M. Cossentino, L. Sabatucci, S. Sorace, and A. Chella. Patterns reuse in the PASSI methodology. In *ESAW-03*, pages 29–31. Springer, 2003.

[13] K. H. Dam and M. Winikoff. Comparing Agent-Oriented Methodologies. In *Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2003)*, volume 14. Springer, 2003.

[14] FIPA. Methodology technical committee. http://www.fipa.org/activities/methodology.html,2003

[15] FIPA Methodology Technical Committee. FIPA-SPEM. http://www.pa.icar.cnr.it/%7ecossentino/FIPAmeth/metamodel.htm

[16] N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

[17] F. Mamei and F. Zambonelli. Programming stigmergic coordination with the TOTA middleware. In *Proceedings of the $4^{th}$ international conference on Autonomous Agents and Multi-Agent Systems, New York (USA)*, pages 415–422, 2005.

[18] A. Molesini, A. Omicini, E. Denti, and A. Ricci. SODA: A roadmap to artefacts. *Engineering Societies in the Agents World VI*, 3963:49–62, 2005.

[19] Object Management Group. SPEM. http://www.omg.org/technology/documents/formal/spem.htm

[20] A. Omicini. SODA: Societies and infrastructures in the analysis and design of agent-based systems. In P. Ciancarini and M. J. Wooldridge, editors, *Agent-Oriented Software Engineering*, volume 1957 of *LNCS*, pages 185–193. Springer, 2001.

[21] A. Omicini and F. Zambonelli. Coordination for internet application development. *Autonomous Anents and Multi-Agent Systems*, 2(3):251–269, 1999.

[22] L. Padgham, M. Winikoff, and A. Melbourne. *The Prometheus Methodology*, pages 217–234. Springer, 2004.

[23] M. Puviani, G. Cabri, and L. Leonardi. Agent Roles: from Methodologies to Infrastructures. In *Proceedings of the 2008 workshop on Role-Based Collaboration, at the 2008 International Symposium on Collaborative Technologies and Systems (CTS'08), Irvine, California, USA, May 2008*, 2008.

[24] Mariachiara Puviani, Giacomo Cabri, and Letizia Leonardi. Agent Roles: from Methodologies to Infrastructures. In *The 2008 workshop on Role-Based Collaboration, at the 2008 International Symposium on Collaborative Technologies and Systems (CTS'08), Irvine California, USA, May 2008*, 05 2008.

[25] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. CArtAgO: A framework for prototyping artifact-based environments in MAS. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for MultiAgent Systems*, volume 4389 of *LNAI*, pages 67–86. Springer, February 2007.

[26] F. Zambonelli, N.R. Jennings, and M. Wooldridge. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12(3):317–370, 2003.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Adaptation of E xtended Polymorphic Self-Slimming Agent Model Into e-Sourcing Platform

Konrad Fuks, Arkadiusz Kawa, Waldemar Wieczerzycki
Poznan University of Economics, al. Niepodległości 10,
60-967 Poznań, Poland
Email: {konrad.fuks, arkadiusz.kawa, w.wieczerzycki}@ae.poznan.pl

***Abstract*—There are two main contributions of the work presented in this paper. First, extended PSA agent model is described. It is based on two new properties of so called *bootstrap agent*: multiplication and parallelism. Second contribution is application of extended PSA model to e-sourcing platform. This duo (e-sourcing and extended PSA model) shows enterprises can significantly increase resources acquisition and potential suppliers search.**

## I. Introduction

DURING last several years, the lowly, back-end sourcing process has been transformed into a strategic resource. Sourcing is now seen not only as a strategic player in the value chain, but as a major driver in the extended supply chain. To enhance their global operations, firms are seeking dynamic supply chain partnerships to increase the speed and intensity of their response to changes in customer demand and to lower costs and reduce risks.[4]

Traditionally, sourcing involves a number of communication mediums to facilitate all business processes between various parties. These include the use of mail, phone, fax, EDI and, more recently, email and the Internet. The current advances in information and communication technology have revolutionized procurement to turn it into a mechanism that enables diverse and geographically disperse companies to create alliances to meet a new form of Internet-oriented consumer demand.[2] Analysts believe that enormous cost savings and efficiencies can be achieved through the utilization of the electronic sourcing, a component of the B2B model. According to the Prime Consulting Group, global firms are optimistic on the level of savings that can be achieved through full implementation of e-sourcing strategies. The potential for savings is tremendous.[5] For instance, General Electric reports that it believes that the firm has saved over $US 10 billion annually through its e-sourcing activities.[4]

However, there are some limitations of e-sourcing. Firstly, there are thousands globally operating e-marketplaces, each with its own range of transaction mechanisms and additional facilities. The increasing number makes it impossible to browse them all by humans. Secondly, there are also differences between the requirements of a prospective participant for buying purposes and those organizations seeking participation as suppliers.[6] Thirdly, the environment of e-marketplaces is constantly changing—some e-marketplaces occasionally give much better offers than others, growing competitive and demand's changeability make the best offer finding difficult.

Those inconveniences can be satisfied by combining e-marketplace with agent technology.

Agent technology has been one of the most prominent and attractive technologies in computer science in the last decade. Software agents and their standards are being developed all the time by a lot of enterprises. Agents can be useful for e-supply chain configuration and management based on e-markets[1], which is strictly related to three particular properties of software agents. Firstly, agents are autonomous. A user can activate them, leave in the network and disconnect, provided agent mission is well defined. Secondly, such agent can be highly mobile. Agents enable dynamic supply chain configuration and reconfiguration in different environments. Thirdly, agents can be very intelligent, thus they are able to support efficient supply chain configuration. In case of e-procurement, agent technology is able to support efficient e-marketplaces, including partners who offer one another the best cooperation possibilities and conditions at a given time. There is a possibility to create temporal and dynamic supply chains aimed at executing a single transaction. This creates previously unknown opportunities for enterprises which can avoid stable supply chains, built on the basis of long-term contracts with rarely replaced business partners. [7][3]

The structure of the paper is as follows.

Section II presents the PSA agent model which is based on multi-dimensional versioning and segmentation of agent code, as well as on distribution of agent code and data over the network. The role of proxy agent and bootstrap agent is detailed. Finally, two extensions of the primary PSA model are discussed.

Section III shows application of the extended PSA model to e-sourcing platform. At the beginning environment of the approach is described. In the second part of this section features and functions of agents as well as exemplification of the approach are presented.

Section IV concludes the paper and shows the future work.

## II. Extended PSA Agent Model

According to the basic PSA (Polymorphic Self-Slimming Agent) model, an agent is composed of a sort of agent head, called *bootstrap agent*, and agent body.[7] Bootstrap agent is relatively small, thus it can be highly mobile. Its goal is to

978-83-60810-14-9/08/$25.00 © 2008 IEEE

move over the network and to decide whether a newly visited environment is potentially interested, taking into account agent mission defined by the user. If it is, then the bootstrap agent recognizes the specificity of the environment, afterwards it communicates with so called *proxy agent*, residing on the origin computer (i.e. a computer in which agent has been created by the user), asking for sending to the bootstrap agent an appropriate part of the code of agent body. Bootstrap agent communicates directly with the proxy agent using the messages in an agent-communication-language (ACL).[8][9]

It may happen that after some time, if the results of agent (i.e. bootstrap agent extended by agent body) activity are satisfactory, the bootstrap agent again communicates the proxy agent asking for the next part of agent body. This situation will be explained later.

If agent mission in the currently visited environment is completed then the agent body is removed from the environment and the bootstrap agent migrates to a new environment or it returns to the origin computer in order to merge with the proxy agent.

For the sake of platform independence and security, we assume that the bootstrap agent is interpreted by a visited environment. In other words the bootstrap agent is a source code, rather than partially compiled (pre-compiled) code.

There are four basic functions provided by the stationary proxy agent:

- It serves as a communication channel between the user and bootstrap agent: it knows current location of bootstrap agent and can influence the path of its movement, as well as tasks performed by the bootstrap agent.
- It encompasses all variants of the agent code that model the variability of agent behavior. Depending on what environment is currently visited by the bootstrap agent, and according to bootstrap agent demands, proxy agent sends a relevant variant of agent code directly to the bootstrap agent, thus enriching it with the skills required in this new environment and, as a consequence, enabling it to continue the global agent mission. Notice, that code transmission redundancy is avoided, since the unnecessary code (not relevant agent variants) remains together with the stationary component.
- It assembles data items that are sent to it directly from the bootstrap agent, extended by a proper agent variant, which are not useful for a mobile code, while could be interesting to the user. The data assembled is stored in so called *knowledge repository*.
- Whenever required, the proxy agent responds to the user who can ask about mission results and data already collected (e.g. a percentage of data initially required). If the user is satisfied with the amount of data already available, proxy agent presents the data to the user and finishes its execution (together with bootstrap agent).

To summarize the aforementioned discussion, one can easily determine the behavior and functions of a moving component (code). When it migrates to next network node, only bootstrap agent is transmitted, while the agent variant is just automatically removed from the previous node. There is no need to carry it together with the bootstrap agent, since it is highly probable that a new environment requires different agent variant. When migration ends, bootstrap agent checks its specificity, and sends a request for a corresponding agent variant transmission, directly to the proxy agent. When code is completed, the mobile agent component restarts its execution.

During the inspection of consecutive network nodes only the information that enriches the intelligence of a moving agent is integrated with agent bootstrap (thus it can slightly grow over the time). Pieces of information that does not increase the agent intelligence are sent by the mobile component of the agent directly to the stationary part, in order to be stored in the data repository managed by it. This agents feature, namely getting rid of unnecessary data, is called self-slimming.

Now we focus on possibilities of the PSA agent versioning, i.e. on the content of the proxy agent that is always ready to select a relevant piece of agent code according to a demand of the bootstrap agent. We distinguish three orthogonal dimensions of agent versioning:

- agent segmentation,
- environmental versioning,
- platform versioning.

*Agent segmentation.* Typically agent mission can be achieved by performing a sequence of relatively autonomous tasks (or stages). Thus the agent code is divided into so called segments corresponding to consecutive tasks that have to be realized. If one task is finished successfully, then the next one can be initiated. Thus, next segment of agent code is received from the proxy agent and agent execution switches to this new segment. Depending on whether agent behavior is sequential or iterative, the previous segment is automatically deleted (in the former case) or it remains in the execution environment (in the latter case).

To summarize, this versioning dimension, namely segmentation, models multi-stage nature of PSA agents.

*Environmental versioning.* Bootstrap agent can visit different environments providing different services, e.g. e-marketplaces, auction services. Moreover, every service can be implemented in a different way. For example, the e-marketplace can be implemented as a web-site or database application. Thus, depending on the specificity of environment being visited different version of agent segment is required.

In other words, the proxy agent keeps and manages sets of versions for each agent segment and takes care on their consistency, i.e. knows which versions can "go together".

To summarize, this versioning dimension, namely environmental, models polymorphic nature of PSA agents.

*Platform versioning.* Finally, every version of every agent segment is available in potentially many variants which are implemented for a particular target environment (i.e. for a particular hardware which runs agent environment: processor, operating system, network communication protocols etc.). There is one especial variant for each agent segment version, that is in a source form. It is sent to the environments which for the security reasons do not accept pre-compiled code. Besides this particular variant, the proxy agent keeps potentially unlimited number of partially compiled variants. If the environment accepts this sort of code, then the proxy agent delivers a variant matching hardware and system software parameters of this environment.

To summarize, this versioning dimension, namely platform, models platform independent nature of PSA agents.

Now we present two important extensions of the primary PSA model, which in our opinion, could be very useful in e-commerce applications (cf. section 3).

Firstly, it may happen that the chain of environments (nodes) visited and examined by a bootstrap agent is very long, before the relevant one is found. In order to increase the efficiency of examination, we allow for a single PSA software agent to have more than one copy of a bootstrap agent, behaving in the same way. In other words, it is possible for the proxy agent to send in parallel, say, *n* bootstrap agents addressed to different network locations. When allowed by visiting environments, they start to examine their specificity in asynchronous way. Their common goal is to find, as fast as possible, the most appropriate environment (e.g. e-marketplace) to fulfill the agent mission.

When all *n* bootstrap agents report the end of their activity and send back results, now it is up to the proxy agent to decide which one of the visited nodes will be examined by the PSA agent (i.e. consecutive variants of code segments). Then, only one of bootstrap agents (augmented by a code delivered to it on demand) continues its execution, while all the remaining *n-1* bootstrap agents automatically "dye" without further actions.

Secondly, we must allow the situation when more than one bootstrap agent reports to the proxy agent a success of examination, i.e. more than one potentially interesting network sites have been found. If they are just e-marketplaces, then perhaps instead of short visiting, they require continuous monitoring of announced offers, 24 hours per day in a long time period. To correctly deal with this situation, we assume that a single PSA agent may have many, so called *twin instances*, residing in different environments. They are fed with necessary code in-the-fly, by the same proxy agent, since they are just "twins" acting in a corresponding way, according to the same code repository kept by a single, shared proxy agent.

Twin instances are to some extent autonomous, thus in general they may work asynchronously. In this case the common proxy agent plays the role of a pure communication mean between them. It may happen, however, that there is a need for synchronicity between twin instances. For example, one instance can switch into next, say, third code segment only if the other one has already finished successfully the execution of first segment. In this case, bootstrap agent is responsible for coordination of instances execution, providing basic synchronization mechanisms.

## III. Application of Extended PSA Agent Model to E-Sourcing Platform

Information systems that support e- sourcing can be classified into four major segments: buy-side applications, sell-side applications, content applications and e-marketplace applications.[4]

In this section we focus on the last segment and we present the adaptation of Polymorphic Self-Slimming Agent Model into it. This new proposal expands the present opportunities offered by e-marketplaces.

Environment of the new approach

- The foundation of resources/suppliers searching is the *sourcing cluster*.
- *Sourcing cluster* (type of business cluster) is a group of enterprises which are looking for the same type of resources (e.g. steel, plastic, packaging, transportation, etc.) created within specific e-marketplace. All sourcing cluster data is stored within e-marketplace database.
- E-marketplaces are based on service-oriented architecture (SOA) which provides system interoperability.
- E-marketplaces coopetite[1] with each other in order to provide wider offers range for their customers.
- Each e-marketplace can have its own company priority managing mechanism. Each company has its own priority level participating on every e-marketplace ( *p-level* ) which can depend on: purchase/ sell volume, purchase/sell value, length of e-marketplace participation period, company size, annual income, country of origin, etc. Characteristics of the prioritization mechanism can be individually set on each e-marketplace.
- E-sourcing platform is a set of all potential interoperable e-marketplaces that can exchange information. Each e-marketplace receives interoperability level status. We distinguished three levels: full interoperability (F), quasi interoperability (Q), base interoperability (B).
- *F status* of e-marketplace must provide: offer browsing, *sourcing cluster* creation and management, agent negotiation, contract signing, partner evaluating (p-level changing), message exchange, and external systems (i.e. ERP, WMS) integration.
- *Q status* of e-marketplace must provide: offer browsing (obligatory), any combination of other *F status* features.
- *B status* of e-marketplace must provide: offer browsing.
- Each enterprise is represented by group of software agents (cf. section II).
- Only enterprises (their software agent representatives) can create or join sourcing clusters.
- Type of the resource in each sourcing cluster is the same.
- The more enterprises in sourcing cluster the bigger possibility to find needed resources.
- At the same time each e-marketplace is searched

[1] Coopetition is a neologism which matches cooperating and competition. Examples of coopetition include Apple and Microsoft building closer ties on software development and the cooperation between Peugeot and Toyota on a new city car for Europe in 2005.

only by one enterprise representative (software agent) from each sourcing cluster.

- E-marketplaces can be accessed by an arbitrary number of software agents (e.g. representing many suppliers) which work independently, or collaborate with other software agents.[3]
- The facilitates offered by the e-marketplace may extend to full execution, including financial and logistical services. [6]

Features and functions of agents

The main c haracteristics of the agents used in our approach are:

- robust ness (even when one or more individuals fail, the group of agents can still perform its tasks),
- self-organization (agents need relatively little supervision or top-down control because they are autonomous; agents have capabilities of task selection, prioritization, goal-directed behavior, decision-making without human intervention)
- flexib ility (agents can quickly adapt to a changing environment because they are very intelligent and reasoning).

Additionally, agents are mobile, so they can relocate their execution onto different places in the network.

We distinguished some groups of features and functions which are characteristic for a proxy and bootstrap agents active on e-sourcing platform.

- Features and functions of proxy agent:
    - It represents particular entities (e.g. purchaser, supplier, logistics service provider) which operate on e-marketplace.
    - It communicates not only with bootstrap agents, but with delegating users (human or IT system of enterprise (e.g. ERP system)).
    - It acts according to guidelines and entitlement granted by a delegating user.
    - It informs about the progress and the results of performed tasks.
    - It solves problems that appear during offers searching.
    - It co opetites with agents from competitive companies.
- Features and functions of bootstrap agent
    - It browses offers available at e-marketplaces.
    - It is responsible for checking and comparing offers of products and services.
    - It notifies proxy agent when a new offer appears and informs about the progress of the realization of the offer.
    - It is responsible for negotiation of terms of cooperation with the agent presenting the offer.
    - It is engaged in business contract monitoring and finally signing it.
    - It is obliged to inform proxy agent about problems and failures.

Exemplification of the approach

Let 's imagine that one company is not satisfied with its past relations with suppliers. Additionally company' transaction volume is still rising and customers want more customized products. Problems with suppliers mainly relate to shipment delays and rigid volumes of orders. This situation forces company to look for new partners to preserve its market position. Task of finding new, reliable suppliers is very difficult and time consuming, especially when company acts alone. Combination of e-sourcing and agent technology (extended PSA model) can inconceivably boost up this process. To start looking for new suppliers our company creates profile on one of available e-marketplaces. Each profile on our future e-marketplace is equipped with group of PSA agents (cf. section II). Then company finds a sourcing cluster on the e-marketplace. If there is no such cluster company creates one, but it still acts alone in searching potential suppliers. When company wants to be part of a bigger sourcing cluster it delegates proxy agent to find one on other e-marketplaces. Proxy agents send bootstrap agents to look through the e-sourcing platform (all e-marketplaces) to find proper sourcing cluster for the company. Company can determine conditions that sourcing cluster must meet (i.e. number of enterprises, size of enterprises, their locations, etc.). Of course for bigger scope of potential suppliers searching, company can participate in many clusters on many e-marketplaces.

But how these sourcing clusters help to find suppliers? Each company in sourcing cluster has its own proxy agent (company representative) and related group of bootstrap agents. Proxy agents of each company associated in the sourcing cluster delegate its bootstrap agents to search for precisely described resources (quantity, quality, price, shipping terms, etc.). Searching conditions ( *Cn* ) are determined individually by each company. *Cn* is a set of *i* conditions. $Cn = \{Cn_1,...,Cn_i\}$. Individual condition ( $Cn_i$ ) can have precise value (number or text) or can be described by text list or number range. When in *Cn* are conditions that can be negotiated or omitted proxy agents creates additional *Cnx* set of information where $Cnx_i$ represents possible state of condition *i*. Each $Cnx_i$ can have one of four states: *ob* for obligatory, *om* for omitted, *n* for negotiable and *n/om* for negotiable or omitted. The last state allows proxy agent to decide what to do with specific condition when resource is founded. Additionally *Cn* set is supplemented with the value of company's total demand for the resource ( *D* ).

Proxy agent delegates bootstrap agents to look for particular resource that meets *Cn* and satisfies *D*. We distinguished four possible scenarios of the resource searching process:

1. All *Cn* are met and *D* can be fully satisfied.
2. All *Cn* are met but *D* can be partially satisfied.
3. Not all *Cn* are met but *D* can be fully satisfied.
4. Not all *Cn* are met and *D* can be partially satisfied.

In cases (1) and (2) proxy agent just sends additional parts of the bootstrap agent source code that is responsible for next sourcing phase (i.e. contract signing). Scenarios (3) and (4) allow proxy agent to change not met $Cn_i$, to start negotiation phase or just to discard the supply offer. All above operations depend on e-marketplace status ( *F*, *Q* or *B* ). When proper features are supported by e-marketplace bootstrap agent source code can be extended with: contract signing procedures, negotiation procedures or message exchange procedures. If *D* is fully satisfied proxy agent informs other proxy agents within its sourcing cluster about founded resource. Information includes basic conditions for the purchase (i.e. supplier location, resource description, shipping information, currency, needed documents, etc.) and updated volume of the resource. All company-specific conditions are omitted in this information.

Above exemplification contains only mechanism for resource searching on external to sourcing cluster e-marketplaces. Implementation of the mechanism and its extension to purchasing, negotiation and contract signing are foundations for future work (cf. section IV).

## IV. Conclusions and Future Work

To summarize, research results reported in this paper mainly relate to well-known li mitations of e-sourcing, i.e. difficulties and restrictions in browsing e-marketplaces by humans, differences between requirements of a potential participant for buying purposes and those organizations seeking participation as suppliers, growing competitive and demands changeability at e-marketplaces that make the best offer finding difficult.

As proposed and explained in the paper, those inconveniences can be solved by software platforms implemented according to very promising and challenging for business – agent technology.

Future work will concern mainly further extensions of PSA model (cf. section II) towards its application in the area of widely understood e-sourcing activities, as well as prototyping experiences, based on the assumptions referred in the previous section (cf. section III). Authors find it also very interesting extending presented model with swarm intelligence algorithms (i.e. ant colony optimization (ACO), particle swarm optimization (PSO)).

## References

[1]. Denkena B., Zwick M., Woelk P.O., *Multiagent-Based process Planning and Scheduling in Context of Supply Chains*, 1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2003, Springer-Verlag, LNAI 2003.

[2]. Folinas, V. Manthau, M. Sigala, M. Vlachopoulou, *E-volution of a supply chain: cases and best practices*, Internet Research, vol. 14, no. 4, 2004.

[3]. Fuks K., Kawa A., Wieczerzycki W., *Dynamic Configuration and Management of e-Supply Chains Based on Internet Public Registries Visited by Clusters of Software Agents*, 3rd International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Springer-Verlag, LNAI 2007.

[4]. Hawking P., Stein A., Wyld D. C., Foster S., *E-Procurement: Is the Ugly Duckling Actually a Swan Down Under?,* Asia Pacific Journal of Marketing and Logistics, Vol. 16, No. 1, 2004.

[5]. Prime Consulting Group, *Reverse Auctions: An Industry White Paper,* International Housewares Association Reverse Auction Task Force, 2002.

[6]. Stockdale R., Standing C., *A framework for the selection of electronic marketplaces: a content analysis approach*, Internet Research: Electronic Networking Applications and Policy, Vol. 12, No. 3, 2002.

[7]. Wieczerzycki W., *Polymorphic Agent Clusters – The Concept to Design Multi-agent Environments Supporting Business Activities*, 2nd International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005, Springer-Verlag, LNAI 2005.

[8]. http://www.fipa.org/specs/fipaSC00001L/, *Foundation for Intelligent Physical Agents*, FIPA Abstract Architecture Specification.

[9]. http://www.fipa.org/specs/fipa00061/, *Foundation for Intelligent Physical Agents*, FIPA ACL Message Structure Specification.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# The Triple Model: Combining Cutting-Edge Web Technologies with a Cognitive Model in an ECA

Maurice Grinberg and Stefan Kostadinov
Central and Eastern European Center for Cognitive Science, New Bulgarian University,
Montevideo 21, 1618 Sofia Bulgaria
Email: mgrinberg@nbu.bg, stefan@yobul.com

***Abstract*—This paper introduces a new model which is intended to combine the power of a connectionist engine based on fast matrix calculation, RDF based memory and inference mechanisms, and affective computing in a hybrid cognitive model. The model is called Triple and has the following three parts: a reasoning module making advantage of RDF based long-term memory by performing fast inferences, a fast connectionist mapping engine, which can establish relevance and similarities, including analogies, and an emotional module which modulates the functioning of the model. The reasoning engine synchronizes the connectionist and the emotional modules which run in parallel, and controls the communication with the user, retrieval from memory, transfer of knowledge, and action execution. The most important cognitive aspects of the model are context sensitivity, specific experiential episodic knowledge and learning. At the same time, the model provides mechanisms of selective attention and action based on anticipation by analogy. The inference and the connectionist modules can be optimized for high performance and thus ensure the real-time usage of the model in agent platforms supporting embodied conversational agents.**

## I Introduction

INTERNET has become an extremely rich environment, which starts to become comparable with a real life environment – the available information is too abundant and the existing options are too numerous to be considered and taken into account completely. Even in the case, in which we can make all possible inferences based on a large ontology in extremely short time (of the order of milliseconds), the increase of information seems to be too large to be useful. It seems that although there is some control in the way the information is presented, namely sometimes in a structured form, pure AI approaches are not sufficient to design and implement artificial cognitive systems in Internet that can achieve human level of performance. Analogously to robotics, accomplishing tasks in complex environments seems to require novel (with respect to GOFAI) approaches starting from the connectionist modeling and going to more and more biologically inspired architectures. Similarly, inference based on ontologies can be done extremely efficiently but this is a problem in itself because of the risks of information explosion.

In cognitive systems this problem is addressed first of all by introducing the concept of working memory (WM) which broadly speaking can represent the most relevant part of the long term memory (LTM) which contains all the knowledge an agent knows. The relevance of the information retrieved from LTM can be insured by activation spreading mechanisms, with the goal and the perceived input (e.g. task, scene etc.) as a source of activation (e.g. see [1]). Such activation spreading can use the connections in LTM or some other types of connection – associative, based on semantical similarity, etc. Other possible mechanisms of selectivity of the information used can be based on anticipatory mechanisms in perception and action (e. g. see [2]—[4]) or other selective attention mechanisms. The role of emotions in cognitive modeling and especially in embodied conversational agents (ECA) has been given a lot of attention recently (e.g. see [5]) and seems to be an important set of mechanisms which can add a lot in communication with users but also influence the reasoning processes. In the case of an ECA which is supposed to interact with users in real time, the problems of scalability and processing time become even more important and efficiency issues are of primary concern.

The model Triple, introduced here for the first time, is aimed at being a cognitive model for cognitive systems and particularly for ECA platforms. It includes, on one hand, several of the necessary mechanisms mentioned above and

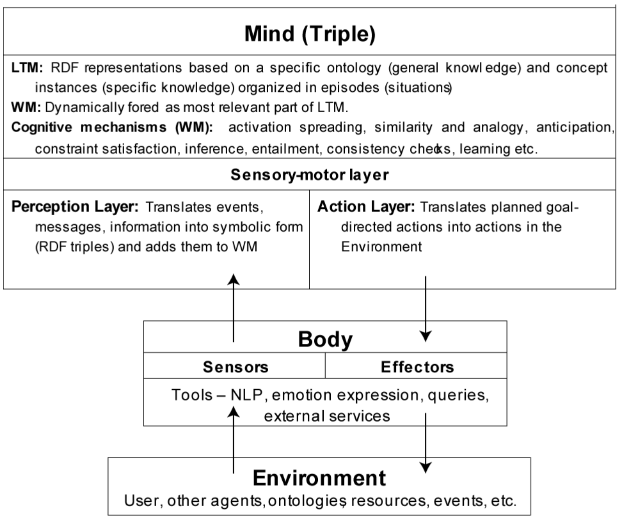

Mind (Triple)
LTM: RDF representations based on a specific ontology (general knowl edge) and concept instances (specific knowledge) organized in episodes (situations)
WM: Dynamically fored as most relevant part of LTM.
Cognitive mechanisms (WM): activation spreading, similarity and analogy, anticipation, constraint satisfaction, inference, entailment, consistency checks, learning etc.
Sensory-motor layer
Perception Layer: Translates events, messages, information into symbolic form (RDF triples) and adds them to WM
Action Layer: Translates planned goal-directed actions into actions in the Environment
Body
Sensors
Effectors
Tools – NLP, emotion expression, queries, external services
Environment
User, other agents, ontologies, resources, events, etc.

Fig 1. A Mind-Body-Environment conceptualization of an ECA, using the Triple model as Mind.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

on the other it tries to achieve maximal computational efficiency in order to allow real time functioning of the ECA. These two constraints lie at the basis of this model: adding all the useful cognitive modeling techniques which allow flexibility, context sensitivity and selectivity of the agent and in the same time – maximal computational optimization of the code and use of very efficient inference methods (e.g. see [6]).

Following this strategy, the model has been designed in three parts that function in parallel. The so called Reasoning Engine (RE) is coordinating and synchronizing the activities of the model and relates the agent with the environment (e. g. user, other agents, etc.) and with the tools the agent can use (e. g. tools to communicate with the user, make actions like access ontologies and data bases, search the Internet (see [3] and [4]), extract LSA information from documents, etc.). RE is also responsible for instance learning – storing of useful episodes in LTM after evaluation. The Inference Engine (IE) is used by RE and can operate on demand. Its main role is to augment parts of WM with inferred knowledge and do consistency checks.

The second part of Triple is the so-called Similarity Assessment Engine (SAE). It is designed to be a connectionist engine, based on fast matrix operations and is supposed to run all the time as an independent parallel process. The main mechanism is activation spreading in combination with additional mechanisms which allow to retrieve knowledge relevant to the task at hand. Communication of SAE with RE is elicited by events related to the level of activation. The SAE mechanisms are supposed to lead to the retrieval of information which is related to the task at hand at different level of abstraction.

The third part is the Emotion Engine (EE) which is based on the FAtiMA emotional agent architecture [7, 8]. FatiMA generates emotions from a subjective appraisal of events and is based on the OCC cognitive theory of emotions [9]. EE, similarly to SAE, is supposed to run in parallel and influence various parameters of the model like the volume of WM, the speed of processing, etc. (see [10] for a possible role of emotions in analogy making). In the same time it will allow achieving higher believability and usability based on the emotional expressions corresponding to the current emotional state of the agent.

The Triple model is connected to the DUAL/AMBR model [1] by inheriting some important mechanisms. Triple like DUAL/AMBR makes use of spreading of activation as a method for retrieval from memory of the most relevant episodic or general knowledge. The mapping of the knowledge retrieved to the task at hand and to the current input to the system is based on similarity and analogy in both models. However the underlying mechanisms are essentially different. In DUAL/AMBR knowledge representation is based on a large number of micro-agents which perform local, decentralized operations and are dualistic in the sense that they spread activation and perform symbolic operations at the same time. The messages exchanged between the micro-agents trigger the establishment of mappings, structural correspondence assessment (which stand for concepts and relations) and the speed of the symbolic processing of the messages depends on their so-called 'energy' (basically an integral over the recent activation). In Triple, an attempt has been made to achieve the same functionality on the basis of clearly separated symbolic mechanisms (reasoning, inference, consistency checks, anticipation, etc.) and connectionist mechanisms (spreading of activation over different types of connections, similarity assessment, distributed representations, etc.). A third component is the emotional module (EE) which is missing in DUAL/AMBR [10]. In DUAL/AMBR the 'duality' is achieved at the level of each micro-agent while in Triple it is achieved by two systems which run in parallel and communicate on event-driven basis. An important additional difference is that Triple is using a full fledged reasoning part in the standard AI sense, which is not available in DUAL/AMBR. The inference and entailment capabilities are integrated with the spreading of activation and evaluation of retrieval and action planning. Only the most active part of WM, corresponding to the focus of attention of the system is subject to augmentation based on inference and to other symbolic processing like evaluation, transfer, and action. The Amine platform [11] has similar augmentation mechanisms which are based on purely symbolic manipulation and are not conditioned by the attention of the system (see [11] and [12] for similar mechanisms like 'elicitation' and 'elaboration').

At this stage Triple inherits from DUAL/AMBR all the mechanisms of transfer of knowledge and anticipation from LTM based on analogy-like reasoning (e.g. see [3] and [4]) but additionally uses consistency checks and inference. The latter is expected to improve efficiency considerably because it will allow timely canceling of impossible plans and will not rely only on external feed back.

One of the main roles of an ECA is to be a sophisticated intelligent interface between a human and a virtual or real (artificial or natural) environments (e.g. see [13] and [14]). An example of such an environments is the Internet, as discussed above. In order to think of the agent as embodied and situated in any environment, the structure shown in Fig 1 has been adopted. The advantage of this representation is the possibility to consider physically and virtually embodied agents on equal footing. It supports also the conceptualization of Internet (or any virtual environment) made above which implies that sufficiently rich virtual environments should be considered as 'real' ones with respect to their richness and complexity. In Fig 1, the Mind is shown with its specific knowledge structures and tasks. The Sensory-Motor Layer makes a mediated connection between the Mind, and the Tools of the agent, which are the sensors and effectors that work with the Sensory-Motor Layer and perceive or act on the Environment. The Sensory-Motor Layer provides symbolically represented knowledge to the Mind and thus makes mediated connection between the Mind and the Environment.

The user occupies the central part in a ECA environment and has a complex set of goals, needs, interests, expectations and previous knowledge. On the other hand, as stressed for instance in [5], any interaction with the user has its affective and emotional background which is highly user and situation specific. In order for such an interface to be maximally efficient under these conditions it has to be highly personalized and context sensitive. It seems that this cannot be done only on the basis of statistical information gathering (e.g. like in a recommender system). The agent has to be a real personalized partner which remembers important (or all) past episodes of interaction with the user and thus is able to establish a deeper relationship with the her based on personalized common experience. For instance, in these memories, the satisfaction of the user should be encoded as, generally speaking, the agent is supposed to satisfy the user. This personalization is achieved in Triple by the rich episodic memory in which all kinds of episodes and information are stored: general knowledge, typical and specific situations, events and user evaluations of task completion outcomes. Examples of how such episodic knowledge can be used can be found in [3] and [4]. Episodes are a rich source of personalized experience and can be used for analogical reasoning or case-based reasoning. They represent a rich source for user-centred learning and generalization. They can encode, among other things, the preferences of the latter in various contexts, his/her definitions of key concepts, etc. and thus allow for information selection and form of presentation.

As our previous experience has shown [4], the success of such an approach is based on the richness of episodic knowledge and in the way the LTM part relevant to the task is accessed. The encoding of episodes is unavoidably specific and unique (a specific set of concept and relation instances is used). The specificity is guaranteed by the learning mechanism consisting in the storage (retention) of task completion episodes and knowledge provided by the user in LTM. It may turn out that the questions or goals are formulated differently than the episodes stored in LTM although using the same concepts, e.g. by using different relations between the same instances of concepts. In order to make use of previously encoded knowledge one needs mechanisms generating various equivalent representations to overcome the limitations of a specific encoding (e.g. the possibility to transform “John is the father of Mary” into “Mary is the daughter of John”). It is obvious that flexibility and reliability in this direction can be achieved only if the similarity and equivalence of knowledge could be assessed, as well as its consistency with previous knowledge. In order to be able to do so, the ECA should dispose of powerful inference and reasoning capabilities in order to achieve maximal flexibility. Thus in comparison to DUAL/AMBR, additionally to analogy making, the Triple model extensively uses rule-based inference and entailment based on previous knowledge based on its relevance.

In the following sections, the main principles of Triple will be presented in more detail and the implementation of the main modules discussed.

## II The Model Implementation

The model presented here is a continuation of an effort to design an ECA architecture with cognitively based mind for a real application [15]. The attempt to use the DUAL/AMBR architecture [1] made evident the existence of scalability problems and the need for faster and more flexible model mechanisms. This lead to the creation of the model Triple.

The embedding of Triple in a real agent platform is currently in progress but some simple evaluations have been already performed with partial functionality and showed promising results in terms of efficiency. The platform implementation (see [15] for details) uses the Nebula engine [16] for the multi-modal generation and NLP processing and speech synthesis [17, 18]. In order to provide the required speed of processing for a real-time application, fast OWLIM RDF inferencing and handling was provided [6] and adapted to the model needs. RDF triples representations and inference on them play a crucial role in the implementation of LTM and IE. All this, combined with the fast matrix-manipulation-based similarity assessment engine provides response time of a few seconds for simple tasks as estimated with the initial implementation of the model.

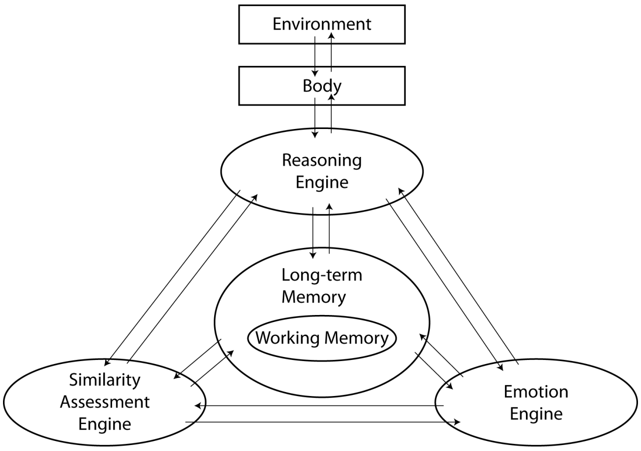


Fig 2. The Triple architecture and interactions among the IE, SAE and RE. (See the text for explanations.)

It should be stressed that the model is intended to be general enough and is not related in an essential way to the use of the specific tools quoted above.

A typical interaction episode with an user would be similar to the interaction episodes described in [4], where DUAL/AMBR has been used as a cognitive model. The user asks a question which is processed by NLP tool [17] and the information about the type of utterance (greeting, question, task, etc.), what is asked for (if the utterance is a question), and what is the knowledge contained in the utterance is expressed in RDF triples and attached to WM as instances of existing in LTM shared concepts. The set of nodes represents the goal or the task given by the user to the agent and is labeled as 'target'. The target could include any additional input accessible to the agent via its 'sensors'. The basic process of task completion is to augment this target set of nodes with knowledge from LTM or from external ontologies or sources and formulate an answer or reaction to the user utterance. The following subsections explain in detail how this is achieved and what is the interplay between the three processing modules of Triple presented schematically in Fig 2.

### A. Similarity Assessment Engine (SAE)

As discussed in previous sections, the agent's knowledge is represented as concepts (including relations) and their instances by nodes related through weighted links forming a semantic network which could include other types of connection (e.g. associative ones) especially in the episodic part (see [1]). The main problem to be solved by SAE is to assess the level of similarity or correspondence between two nodes typically belonging to the target (the task) of the agent and to LTM, respectively. The problem with similarity assessment is central for such approaches and has been extensively explored in the analogy and case-based reasoning literature (e.g. see [19, 20, 21] and the references there in). The Similarity Assessment Engine (SAE) is exploring some established and some new approaches in finding similarities aiming at the highest possible efficiency.

Typically, part of the nodes come from the target (e.g. a task by the user or an internal goal for the agent) and the others come from the active part of LTM and both form the WM content. The normalized similarity (lying between 0 and 1) between target and LTM nodes is taken to be the probability of correspondence between the two nodes. The basis for similarity evaluation can be quite various and generally speaking takes into account the taxonomic links and the proximity of the nodes in the semantic tree by using different types of distributed representations, some of which will be explained bellow. Additional mechanisms as the one proposed in [19] using Latent Semantic Analysis (LSA) [22, 23] and are based on semantic similarity.

The connectionist processing is based on the representation of the WM of the agent as a set of matrices, that reflect different dependencies among the concepts, relations and their instances. It should be stressed that there are no named connections in this representation and all relations including 'instance-of', 'sub-class', 'super-class' and others are represented as nodes, linked by weighted connections. Thus all the contents of the WM are represented as a vector of nodes (concepts, relations and their instances) and the connections are a matrix of weights. All the nodes related to the input and the goal for the agent are part of a 'target' set of nodes and the achievement of the goal (e. g. finding the answer of a question) is formed by augmenting this target set so that it contains the results of the processing and include completion of the goal or a failure. In order to do so the agent must retrieve knowledge similar (or analogical) to the input and transfer the required knowledge to the target or trigger a plan of actions in order to find it, e. g. search in an ontology or data source. So, to each node in WM a vector is attached, which contains distributed information about its definition, relations and actions in which it participates, its instances or super-classes, etc. which can be used to measure its similarity to other nodes. It should be noted that this distributed representation is spanned only over the nodes in WM, i. e. the nodes with sufficient activation. Moreover, each term in the distributed representation is multiplied by the corresponding node activation. This procedure will change the similarity between nodes depending on the activities of the nodes present in the WM. Thus similarity becomes dynamic and changes depending on the content of WM at a given time and thus is highly context dependent. For example if the neighboring nodes of a node have a higher activation than the higher concepts in the taxonomy more specific similarity will be found and if the higher concepts are with larger activation more abstract and high level similarity (analogy) will be established. The similarity depends on a similarity measure between any two such characteristic vectors (e.g. a normalized scalar product). This normalized similarity is interpreted in the model as the probability for useful correspondence between the two nodes which eventually will lead to the retrieval and mapping of the target task to a relevant part in LTM, which may contain the required knowledge for task completion.

This principle of similarity assessment seems to be quite general and we are planing to test and combine different implementations. All are based on the relations between the nodes in Triple LTM. Other methods we are considering are the ones based on Latency Semantic Analysis [15, 16] for the respective domains. Additional mechanisms, presently explored are related to spreading of different type of activation which require much more space and will be reported in a separate future publication as applied to a music domain. The SAE has been implemented using compiled Matlab code and making use of the sparse matrices manipulation routines.

### B. Emotional Engine(EE)

The Emotional Engine works continuously and in parallel with the SAE and provides continuous fine-tuning of the rest of the Mind modules and, from time to time – takes control over the reasoning process (see end of this section). The so-called fine-tuning is done as the EE is constantly decaying to a “neutral” state. The size of the Working Memory, the severity of the reasoning constraints, etc. all depend on the emotional state. As the emotional state changes over time, when major changes occur, events are sent to the Reasoning Engine. An example of such event is the case where an action of information retrieval from an Internet web-service is performed. If the service does not respond at all, the emotional state would decay over time and thus the action that triggered the emotional state would be suspended. This solution seems better than to have a fixed amount of “real” time (in seconds), as the EE depends on the time elapsed, the importance and desirability of the action performed, the initial emotional state and any other events that might occur in the meantime. When an action is interrupted if proven unsuccessful, or because the EE halted it, the next action in the line is going to be executed (see Reasoning Engine section for details). The details about this engine will be published in a forthcoming paper [8].

### C. Reasoning Engine (RE)

The reasoning engine integrates and controls the operation of the SAE, IE, and EE on one hand and the communication flow with the Body (and thus with the Environment) on the other (see Fig 2). It works with the sensory-motor part of the Mind by receiving the information from the sensors (user utterances, information about results from actions, etc.) and sending action commands to the Environment via the Body's tools (Multi-Modal Generation, Data Source Lookup, etc.). The main interactions with the Body and Environment are the same as the ones reported in [4] and are briefly explained in the next section.

### *I Information flow*

Information (in RDF triple form) received by the Mind is either a new information from the Environment (question, task, definition, etc.), or a response from an action (result from a tool). The human input is considered as information from the Environment and is processed by the NLP tool into a set of RDF triples. Each coalition, coming from the Body, is assigned a context, and if there is no current task, a new context is created.

When there is an action coalition of nodes transferred (see next section), the RE identifies the statements related to this coalition and sends them to the Body as an "action command". In order to be meaningful, this action command must adhere to the requirements of the specific tool it is addressed to.

### *II Main mechanisms*

When a new message is received from the Body, the RE adds it to WM (and thus – to LTM) and marks all the statements from the message as "target". All the parts of the message, that are not internal (e.g. which Tool generated the message) are marked as being "goal" for the system (see [4]). The target set (called "input" and "goal") is the source of activation for the SAE module. This module is started by RE and gives continuously information about similarities found between the target set and knowledge in WM and initially determines the focus of attention of the agent – the most active part in WM. SAE estimates the level of similarity and based on that RE establishes candidate correspondences between the target set and the LTM contents in WM in the so-called Similarity Assessment and Correspondence Processors.

It should be stressed that in all tasks the ultimate goal is to satisfy the user by providing the needed information or solution of the task.

More precisely, when the task is to answer a question this general goal would be to provide the user (or another agent) with the answer. Initially, the goal of the system is quite general but with the processing of the question it becomes more specific. For instance, for the currently implemented music domain [15], the goal could be to give the name of an album of a singer, his/hers religious status, birth date, etc. In the current, early development stage, the architecture does not have an explicit planning mechanisms, although the reinforcement learning (by always aiming at user's satisfaction), top-down learning (episode retention) and action decision are present and should allow the model to find better and better solutions over time and encode sequences of actions into episodes. During the future development, different approaches will be considered for inclusion like BDI features and emotionally guided planning.

As stated in the beginning of this section, messages from the Body are added to the WM and become source of activation for the SAE that starts to send similarity assessments to the RE. Those assessments are checked for obvious flaws and inconsistencies due to the fact that SAE is supposed to make analogies as well. If no flaws are found, they are transformed to established correspondences. When the correspondences between the target set and LTM are established the IE is used to verify and evaluate them and eventually the candidates are rejected or confirmed. Based on the existing correspondences, parts of past episodes are evaluated by the IE and transferred to the target set until eventually an action transfer is chosen and the appropriate action structure is added to the target set.

When an action structure is added to the target episode, it is automatically sent to the Body, along with its canonical representation. Each and every Tool that is going to receive and process the action command expects a specific format, and so a canonical message structure is needed. Those structures are kept in the Mind and are used when an action command is sent. Usually, they contain a sub-graph (always the same) that identifies the type of Tool to process the command, any additional information (as there might be Tools, carrying on various tasks) and the actual command.

Finally the user is provided with the result of the task and could give a feedback. If the task is considered completed (e. g. the question is answered and the user is satisfied with it) the whole episode with the task and its completion is stored in LTM as an experience episode. Any new knowledge acquired in the scope of the current task is isolated as general knowledge. In the current implementation, the user has two buttons in the interface – " praise" and "scold". If there is no button pressed after the answer is provided by the agent, the system assumes the user was satisfied and records the episode as successful. If the user presses the "praise" button, the episode is recorded as "more than successful" and if the user presses the "scold" button, the system records the episode as not-successful. In the latter case, this episode is recorded in the WM with very low chance of being retrieved in the future and as a negative example.

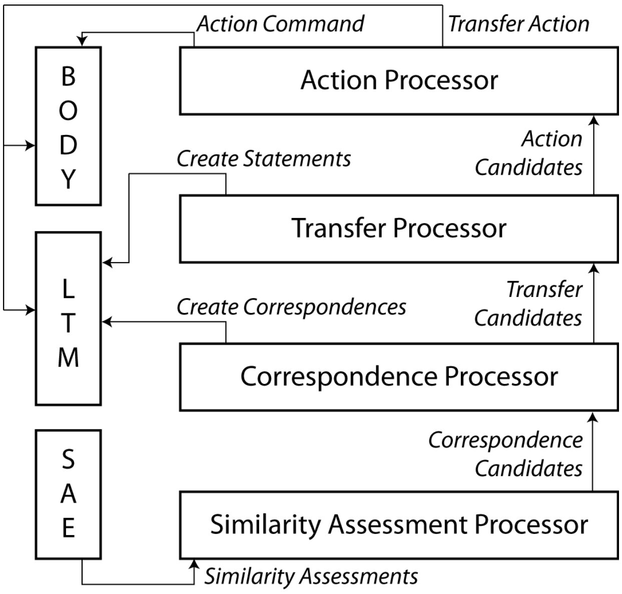


Fig 3: Reasoning Engine Modules.

### *III RE modules*

The RE is made of several procedures for handling incoming knowledge structures and with several modules that are called "processors." The latter are used asynchronously, serially and iteratively within the scope of a single task. They are briefly described here, as the implementation details are not so important for the model and as they are still subject of changes, assessment and fine-tuning.

The modules of RE are as follows:

- Similarity Assessment Processor: handles the initial similarity assessments from SAE, eliminates the inconsistency and establishes correspondence hypotheses (CH). The latter are sent to the Correspondence Processor. If there are no correspondences established, the module aborts execution of the current reasoning process and invokes again the SAE;
- Correspondence Processor: the correspondence hypotheses are processed by this module. Each correspondence hypothesis has a score assigned on the basis of the activation of the nodes it puts into correspondence and of its consistency with other correspondence hypotheses. The most active correspondence hypotheses are used as a base for the formation of a list of transfer requests that are sent to the Transfer Processor. Before this, the correspondence hypotheses are checked against the contents of the LTM (fast, low-cost and efficient process). It is still a point of research if the check against the LTM contents is more efficient if done only in SAP, only in the Correspondence Processor, or in both of them;
- Transfer Processor: removes inconsistent transfer requests, deals with contradictory ones and evaluates them on the basis of the activation of the corresponding nodes and relatedness to the current goal. This is the module that actually adds transferred knowledge to WM. This module also creates a list of action requests, which are sent to the Action Processor;
- Action Processor: receives a list of possible actions to execute from the Transfer Processor. Contradictory actions are evaluated, based on their activation and the one with the highest activation is executed. If the last action has proven unsuccessful (e.g. no answer from an information search tool after a fixed amount of time).

### IV Inference Engine (IE)

The RDF-based agent LTM provides the permanent storage of the agent knowledge. It is updated during operation of the model and can scale up to hundreds of thousands of statements. The current working prototype uses a LTM of less than ten thousand statements. The knowledge is expressed entirely with RDF triples and the model has mediated connection to the environment, as it receives and sends (to the 'Body') coalitions of RDF triples.

As mentioned above, the RDF representation provides the basis for extremely efficient reasoning-based (both by inference and analogy) augmentation of the  goal and the most active part of the WM (which has the focus of attention of the agent). This allows the system to "transfer" specific and relevant information at very high speeds. The inference capabilities and the RDF storage as a whole are also used for verification and consistency check by the memory retrieval and transfer of knowledge for task completion purposes (e.g. answering a user question). The episodes that are stored in the LTM are identified by the contexts of the statements. As one statement can be part of more than one context, there are nesting and overlapping contexts, allowing flexible behavior and less data multiplication.

The IE uses mechanisms which are used by many platforms. However the novelty here is the combination with connectionist activation spreading which selects the knowledge in LTM relevant to the task at hand. This mechanism is inherited from DUAL/AMBR [1] but, as explained bellow, is further developed to include an attentional focus and make specific use of inference based memory augmentation.

## V Conclusion

In this paper, the progress in the development of a new model for an embodied conversational agent is presented. Only the main ideas behind the model, its basic modules and the interplay among them are presented, although the largest part of the mechanisms have been implemented. The main focus was to outline the architecture of the model, the basic mechanisms allowing to increase its efficiency and the role of the reasoning engine in the interplay between the engines.

The main idea behind this model is to combine a rich cognitive model which would bring flexibility, context sensitivity, and selective attentional mechanisms with cutting edge fast machine learning algorithms like logical inference over ontologies and fast matrix calculations. At the core of the model is the combination of an connectionist similarity assessment and emotion modules which run continuously in parallel and a serial reasoning engine, which is based on inference over RDF triples. The main principle behind the connectionist engine is to combine activation spreading and semantic relevance and relational information in order to focus any further operations only on the most useful part of LTM. This fast selection of only a small part the agent's knowledge allow for its further augmentation by efficient inferences.

The three main 'engines' of the model – the reasoning, the emotional, and the similarity assessment engines – have been already implemented and parts of them tested. Although the results are very promising they are too preliminary to be reported here. The elaboration of the model, its full integration within a full fledged agent platform and tests with real users are currently in progress and will be the subject of a next papers. One of the latter will be focused on the detailed presentation and tests of the SAE and the other on the role of the EE as evidenced by simulations and usability tests.

## Acknowledgment

We would like to gratefully acknowledge the discussions with Dr. Hristo Laftchiev, Kiril Kiryazov, Atanas Kyriakov, and Damian Ognianoff about the presented model.
The work is supported by the project RASCALLI under FP6 of EC.

## References

[1] B. Kokinov, "A hybrid model of reasoning by analogy," in K. Holyoak and J. Barnden (Eds.), *Advances in connectionist and neural computation theory: Vol. 2. Analogical connections,* Norwood, NJ: Ablex, 1994, pp. 247 – 18.

[2] K. Kiryazov, G. Petkov, M. Grinberg, B. Kokinov, and C. Balkenius, "The Interplay of Analogy-Making with Active Vision and Motor Control in Anticipatory Robots," *Anticipatory Behavior in Adaptive*

*Learning Systems: From Brains to Individual and Social Behavior*, LNAI 4520, 2007.

[3] S. Kostadinov, G. Petkov, and M. Grinberg, "Embodied conversational agent based on the DUAL cognitive architecture," in *Proc. of WEBIST 2008 International Conference on Web Information Systems and Technologies*, Madeira, Portugal.

[4] S. Kostadinov and M. Grinberg, "The Embodiment of a DUAL/AMBR Based Cognitive Model in the RASCALLI Multi-Agent Platform," in *Proc. 8th International Conference on Intelligent Virtual Agents*, Tokyo, LNCS 5208, 2008, pp. 35–363.

[5] C. Becker, S. Kopp, and I. Wachsmuth, "Why emotions should be integrated into conversational agents," in *T. Nishida (Ed.), Conversational Informatics: An Engineering Approach*, Chichester: John Wiley & Sons, 2007, pp. 49–68.

[6] A. Kiryakov, D. Ognyanoff and D. Manov, " OWLIM – A Pragmatic Semantic Repository for OWL," in *Proc. Information Systems Engineering – WISE 2005 Workshops* , LNCS 3807, pp. 182 – 192 .

[7] J. Dias and A. Paiva, "Feeling and Reasoning: A Computational Model for Emotional Characters," *Progress in Artificial Intelligence*, Berlin, Springer, 2005.

[8] J. Dias, K. Kiryazov, A. Paiva, M. Grinberg, and S. Kostadinov, "Integration of Emotional Mechanisms in the Triple Agent Model,." In preparation.

[9] A. Ortony, G. Clore, and A. Collins, "*The Cognitive Structure of Emotions,"* Cambridge University Press, UK, 1988.

[10] I. Vankov, K. Kiryazov, and M. Grinberg, " Impact of emotions on an analogy-making robot," in *Proceedings of CogSci 2008,* Washington DC, July 22–26.

[11] A. Kabbaj, "Development of Intelligent Systems and Multi-Agents Systems with Amine Platform", in Proc. *ICCS 2006*, LNCS 4068, pp. 286-299.

[12] A. Kabbaj et al., "Ontology in Amine Platform: Structures and Processes", in Proc. *ICCS 2006* , LNCS 4068, pp. 300-313.

[13] J. Cassell, "Embodied conversational agents: representation and intelligence in user interfaces," in *AI Magazine archive. Volume 22, Issue 4,* pp. 67–8, 2001.

[14] N. Le β mann, S. Kopp, and I. Wachsmuth, "Situa ted interaction with a virtual human perception, action, and cognition," in G. Rickheit and I. Wachsmuth (Eds.), Situated Communication, Berlin: Mouton de Gruyter, 2006, pp. 287–323.

[15] B. Krenn, "RASCALLI. Responsive Artificial Situated Cognitive Agents Living and Learning on the Internet", in *Proc. of the International Conference on Cognitive Systems (CogSys 2008)*, LNCS 4131, pp. 535-542.

[16] NEBULA Engine (2007-2008), http://www.radonlabs.de/technologynebula2.html

[17] Feiyu Xu, H. Uszkoreit, Hong Li, "A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity," in *Proc. of ACL 2007,* Prague.

[18] MARY Text-to-Speech Engine (2008) – http://mary.dfki.de .

[19] M. Ramscar and D. Yarlett, "Semantic grounding in models of analogy: an environmental approach," Cognitive Science 27, 2003, pp. 41 – 71.

[20] L. B. Larkey and A. B. Markman, "Processes of similarity judgment", in *Cognitive Science 29, 2005*, pp. 1061–1075.

[21] R. Lopez de Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. Cox, K. Forbus, M. Keane, A. Aamodt, I. Watson, "Retrieval, Reuse, Revision, and Retention in CBR", in *Knowledge Engineering Review, 20(3), 2005*, pp. 215–240.

[22] K. A. Ericsson and W. Kintsch, "Long-term working memory," *PsychologicalReview*, v. 102, 1995, pp. 211 – 245.

[23] W. Kintsch, V. L. Patel and K. A. Ericsson, " The role of long-term working memory in text comprehension," *Psychologia*, v. 42, 1999, pp. 186-198.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 39–45

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Planning and Re-planning in Multi-actors Scenarios by means of Social Commitments

Antonín Komenda, Michal Pěchouček, Jiří Bíba, Jiří Vokřínek
Gerstner Laboratory – Agent Technology Center
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{antonin.komenda, michal.pechoucek, jiri.biba, jiri.vokrinek}@agents.felk.cvut.cz

***Abstract*—We present an approach to plan representation in multi-actors scenarios that is suitable for flexible replanning and plan revision purposes. The key idea of the presented approach is in integration of (i) the results of an arbitrary HTN (hierarchical task network) -oriented planner with (ii) the concept of commitments, as a theoretically studied formalism representing mutual relations among intentions of collaborating agents. The paper presents formal model of recursive form of commitments and discusses how it can be deployed to a selected hierarchical planning scenario[1].**

## I. Introduction

COOPERATION between intelligent agents is usually established by means of negotiation resulting in a set of obligations for the participating agents that lead onwards to achievement of a common goal agreed to by the agents. Wooldridge and Jennings formalize the obligations by describing the cooperative problem solving by means of *social commitments* [1]—the agents commit themselves to carry out actions in the social plan leading onwards to achievement of their joint persistent goal [2].

The problem of distributed planning (DP) has been often discussed in the AI planning and multi-agent research communities recently (e.g. [3], [4], [5], [6]). Distributed planning has been viewed as either (*i*) planning for activities and resources allocated among distributed agents, (*ii*) distributed (parallel) computation aimed at plan construction or (*iii*) plan merging activity. The classical work of Durfee [3] divides the planning process into five separate phases: task decomposition, subtask delegation, conflict detection, individual planning and plan merging.

The distributed planning approach proposed in this paper does not provide constructive algorithms for dealing with either of the phases. Instead we propose a special mechanism for plan execution in distributed, multi-actor environment. As such it will affect all the phases of the Durfee's distributed planning architecture.

While classical planning algorithms produce a series of partially ordered actions to be performed by individual actors, we propose an extension of the product (but also an object) of the planning process so that it provides richer information about the context of execution of the specific action. The context shall be particularly targeted towards mutual relation between the actions to be performed by individual actors and shall be used mainly for replanning and plan repair purposes.

The planning problem we are trying to deal with can be informally understood as the task of solving a classical HTN (hierarchical task network) planning problem, defined by an initial partially ordered (causally connected) series of goals, by a set of admissible operators (defined by their preconditions and effects) and methods suggesting a decomposition of a goal into a lower-level planning problem. The plan can be sought for by an individual actor or in collaboration of multiple actors (sharing knowledge and resources). The product of planning is a set of partially ordered terminal actions, allocated to individual actors who agreed to implement the actions under certain circumstances. These circumstances are expressed by specific commitments including the following pieces of information:

- *commitment condition* that may be (*i*) a specific situation in the environment (such as completion of some precondition) or (*ii*) a time interval in which the action is to be implemented no matter what the status of the environment is or (*iii*) a combination of both.
- *decommitment conditions* specifying under which condition the actor is allowed to recommit from the commitment once the task is finished (e.g. notification) or once the task cannot be completed (e.g. a failure)

For long, multi-agent research community has been providing interesting results in the formal work in the field of agents' social commitment, as specific knowledge structures detailing agents individual and mutual commitments. The presented research builds on and extends this work.

The article is structured as follows. In the section II, the formal description of commitments by Wooldridge is extended, a recurrent notation formalizing the commitments is presented and its use for distributed planning purposes is shown using a scenario for verification. The section III gives a brief overview of the most relevant works to our approach. Finally, the last section concludes the paper.

[1] The presented research has been supported by I-GLOBE (http://i-globe.info/), the US ARMY, CERDEC project (no.: N62558-06-P-0353). The project has been performed in cooperationwith University of Edinburgh, Artificial Intelligence Application Institute

## II. Commitments for Planning and Re-planning

As stated in the introduction, a social commitment is a knowledge structure describing an agent's obligation to achieve a specific goal, if a specific condition is made valid and how it can drop the commitment if it cannot be achieved. The commitment does not capture description how the committed goal can be achieved. Individual planning for a goal achievement, plan execution and monitoring is a subject of agents internal reasoning processes and is not represented in the commitment.

In the context of the planning problem defined in the Introduction, we understand the agent's specific goal (to which it commits) as an individual action, a component of the plan, which resulted from the given planning problem. While typical action in a plan contains only a precondition and an effect, in this paper we will describe how its representation can be extended so that the commitment-related information is included.

Michael Wooldridge in [7] defines the commitments formally as follows:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\ &\quad\lambda=\{(\rho_1,\gamma_1),(\rho_2,\gamma_2),\ldots,(\rho_k,\gamma_k)\},\end{aligned} \tag{1}$$

where $A$ denotes a committing actor, $\psi$ is an activation condition, $\varphi$ is a commitment goal, and $\lambda$ is a convention. The convention is a set of tuples $(\rho,\gamma)$ where $\rho$ is a decommitment condition and $\gamma$ is an inevitable outcome. The convention describes all possible ways how the commitment can be dropped. Generally speaking, the actor $A$ has to transform the world-state in such a way that the $\varphi$ goal becomes true if $\psi$ holds and any $\gamma$ has not been made true yet. The actor is allowed to drop the commitment if and only if $\exists i : \rho_i$ which is valid. A decommitment is allowed provided that $\gamma_i$ is made true. A formal definition in modal logic (working with the models of mental attitudes like Believes, Desires, Intentions, [8], and temporal logic where the operator AG denotes an the inevitability and operator $\curvearrowleft$ denotes the temporal until) follows as defined in [7]:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda)\equiv\\ &\quad((\mathsf{Bel}\ A\ \psi)\Rightarrow \mathsf{AG}((\mathsf{Int}\ A\ \varphi)\\ &\qquad\wedge(((\mathsf{Bel}\ A\ \rho_1)\Rightarrow \mathsf{AG}((\mathsf{Int}\ A\ \gamma_1)))\curvearrowleft\gamma_1)\\ &\qquad\ldots\\ &\qquad\wedge(((\mathsf{Bel}\ A\ \rho_k)\Rightarrow \mathsf{AG}((\mathsf{Int}\ A\ \gamma_k)))\curvearrowleft\gamma_k)\\ &\quad)\curvearrowleft\bigvee_i\gamma_i).\end{aligned} \tag{2}$$

This definition is used in a declarative way. Provided that whatever the agent does during a specific behavior run complies with the above defined commitment, the expression 2 is valid throughout the whole duration of the run.

One of the goals of the research described in this paper was to provide a formalism for networked commitments to be used for replanning. As clearly stated in the introduction, the commitment conditions can represent variable bindings among preconditions and effects of the individual commitments achieved either by monitoring the environment status or by inter-agent communication (e.g. reception of a specific trigger message). Such representation would be very inflexible in practical applications as it would either need the agents to do nothing and wait for an inhibiting event to happen or risk that once an inhibiting event happens the agent will be busy performing other commitments. Therefore the agents may want to engage in booking and the commitment's precondition would contain fixed time when the commitment is supposed to be adopted. The most flexible approach would be a combination of both—inhibition event and preliminary booked time window, specifying when the inhibiting event is likely to happen. Let us assume that this is the case in the remainder of the paper.

In the distributed plan execution a failure may occur. The indirect impact of this failure may be e.g. a situation where the arranged inhibition event will not happen in the preliminary booked time window. Such occurrence may invoke replanning and allow some agents to e.g. drop unnecessary commitments. This is the reason why the commitments shall not be linked one with other not only via preconditions but also by means of variable bindings among individual agent's decommitment rules. Using these bindings, we can describe the causal sequentiality of the commitments and requests for particular decommitments—Fig. 1.

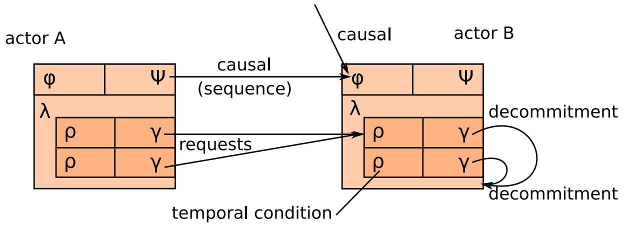


Fig. 1. Commitments and bindings—the actor A's commitment influences the actor B's commitment using the causal (sequential) link, the link is described using the $\psi$ and $\varphi$ clauses (e.g. $\psi = \texttt{building-is-ready(B)}$ and $\varphi = \texttt{ready(B)}$). The actor B's commitment is influenced by external causality too. The actor B's commitment can be decommitted in two cases: either the *temporal condition* $\rho$ becomes true or one of the actor A's rules *requests* the decommitting. The decommitment request is triggered by one of the actor A's $\rho$ conditions.

While we will be generalizing on the process of decommitment later in the paper, let us work for now with the specific particular decommitment case suggested in the previous paragraph. Let us assume one agent $A$ forcing decommitment of the other agent's $B$ commitment by means of setting a value of a variable contained in the other agent's commitment. The agent $A$ contains a commitment with a decommitment rule in the form $\langle\rho, v\rangle$ and the agent $B$ contains a commitment with a decommitment rule in the form $\langle v, \texttt{decommit}(B)\rangle \in \lambda_A$.The request is started by $\rho$ precondition of the actor $A$ (e.g. decommitting the $A$'s commitment). Thus the actor $A$ intends to make the variable $v$ valid. This causes the agent $B$ to intend to decommit by intending the variable decommit(B) to be valid (see Fig. 1).

This clear example uncovers two needed extensions of the classical social commitment model: (*i*) recurrence of the commitment form—enabling a possibility to disable (decommit) a decommitment request and (*ii*) explicit termination condition—describing termination without any intentional part.

### A. *Commitment Recurrence*

The original Wooldridge definition of a commitment makes a clear distinction between the commitment subject ($\varphi$) and the mini-goals set in the commitment convention ($\gamma$). While there is a mechanism for the agent to drop $\varphi$, a once adopted mini-goal $\gamma$ cannot be decommitted. Due to high dynamism and uncertainty of the target scenario, we assume the re-planning and plan repair mechanisms to be substantially more complex. We require that the mechanism would allow the agent to try out several different decommitment alternatives, based on the current properties of the environment. The set $\lambda$, allows listing various different decommitment rules, while no mechanism have been specified how different decommitment alternatives are tried out.

That is why we propose generalization of the commitment so that each goal in the commitment structure can be treated equally. Let us introduce the recursive form of a commitment, which enables the nesting of the commitments—Fig. 2:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda^*), \lambda^* = \\ &\quad\{(\mathsf{Commit}\ x_1\ \rho_1\ \gamma_1\ \lambda_1^*), \\ &\quad(\mathsf{Commit}\ x_2\ \rho_2\ \gamma_2\ \lambda_2^*), \ldots, \\ &\quad(\mathsf{Commit}\ x_k\ \rho_k\ \gamma_k\ \lambda_k^*)\}.\end{aligned} \tag{3}$$

The formula 3 extends the definition in 2 not only by inclusion of a set of decommitment rules in each of the individual decommitment rules. It also allows the newly adopted commitments to be assigned to different actors. The delegation kind of decommitment between two agents $A$ and $B$ would have the following form:

$$(\mathsf{Commit}\ A\ \psi\ \varphi\ \{(\mathsf{Commit}\ B\ \rho\ \varphi\ \emptyset)\}), \tag{4}$$

representing that agent $A$ can drop the commitment towards $\varphi$ provided that $\rho$ is valid and provided that $B$ accepts a commitment towards $\varphi$ on $A$'s behalf.

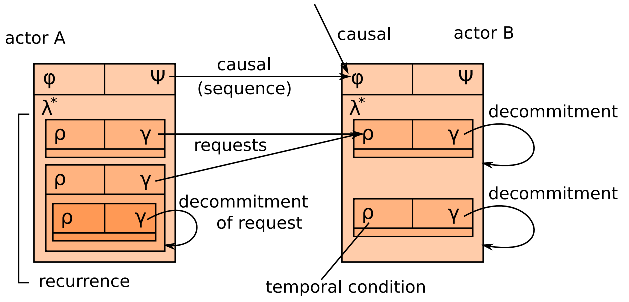


Fig. 2. Commitment and its $\lambda^*$ commitments—the Fig. 1. is extended by one *decommitment of request* which can be decommitted if the most inner $\rho$ condition becomes true. Decommitting of the request causes the actor B's commitment cannot be decommited by the actor A's convention goal any more. Here the recursive form enables the nesting of the inner commitment.

This form is very expressive in the sense of the description of exceptional states. It allows us to have a branched chain of individual nested commitments for each individual situation. The recursive nature allows us to describe an arbitrarily complex protocol using only one knowledge base structure—a recursive form of the commitment. The recursive form of the commitment is thus defined as:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda^*) \equiv \\ &\quad((\mathsf{Bel}\ A\ \psi) \Rightarrow \mathrm{A}((\mathsf{Int}\ A\ \varphi) \wedge \bigwedge_j \lambda_j^*) \curvearrowleft \bigvee_i \gamma_i).\end{aligned} \tag{5}$$

### B. *Termination Condition*

We have explained in Section II that if the agent complies with the commitment, the formula 2 is always valid. However, this implication is not bidirectional. If we use this commitment definition for writing a computer program, running the behavior of an agent, we would need that all the runs that can be implemented by the formula 2 implement agent's correct commitment. In order to do this we need to show how a termination condition can be modeled by means of the social commitment. Let us assume we wanted to implement e.g. the blind commitment. According to [7] the blind commitment is defined as

$$(\mathsf{Commit}\ A\ \varphi) \equiv \mathsf{AG}(\mathsf{Int}\ A\ \varphi) \curvearrowleft (\mathsf{Bel}\ A\ \varphi) \tag{6}$$

Here the term $\mathsf{Bel}(A\ \varphi)$ is the simplest example of a termination condition. The termination condition here would be described using the $\lambda^*$ commitment as follows:

$$(\mathsf{Commit}\ A\ \mathtt{false}\ (\mathsf{Bel}\ A\ \varphi)\ \emptyset). \tag{7}$$

A general termination condition $t$ in the commitment model can be defined as follows:

$$(\mathsf{Commit}\ A\ \mathtt{false}\ t\ \emptyset). \tag{8}$$

The condition ($\mathtt{false}$) will never trigger the intention towards the termination condition—$t$. Termination condition of the rule plays an important role here as it will be added to the *until*-part of the commitment and allows the commitment to be valid even if the intention is dropped provided that the termination condition $t$ is valid. Therefore we can extend the set of decommitment rules with a set of termination conditions $\mathcal{T}$ as follows:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \{\lambda^* \cup \mathcal{T}\}), \mathcal{T} = \{t_1, \ldots, t_k\} \\ &(\mathsf{Commit}\ A\ \psi\ \varphi\ \{\lambda^* \cup \mathcal{T}\}) \equiv \\ &\quad((\mathsf{Bel}\ A\ \psi) \Rightarrow \mathrm{A}((\mathsf{Int}\ A\ \varphi) \wedge \bigwedge_j \lambda_j^*) \curvearrowleft \bigvee_i \gamma_i \bigvee_k t_k).\end{aligned} \tag{9}$$

### C. *Decommitment Rules*

We require the agents that perform intelligent planning and replanning by means of social commitments to be able to perform at least basic reasoning about the decommitment rules attached to the particular commitments. This is needed at the time of replanning, when an agent needs to decide which decommitment rule (i.e. a new commitment) to adopt, provided that conditions for more than one are satisfied. Similarly, agents, when they negotiate about who will accept which commitment, shall be able to analyze not only properties of the goal and costs associated with the goal completion process but also the various decommitment rules when considering likelihood of the particular failure to happen. Ideally, the agent shall be able to estimate costs of each decommitment rule. However, with the lack of information about the dynamics of the environment, we will be only able to partially order the decommitment rules by assigning them to different types. Let us introduce four different types of decommitment rules:

- *Termination conditions* (TC)—as described in the Section II-B. These are obviously the most preferred decommitment rules as no further action is required for dropping the particular commitment.
- *Individual commitments* (IC)—commitments that do not involve other agent than the agent itself. These commitments shall be used if the impact of a failure within the multi-agent community shall be minimized. Individual commitments shall represent several other ways how an agent can accomplish a given task.
- *Delegation* (D)—by using this type of commitments the agent shall be able to find some other agent who will be able to complete its commitment on the original agent's behalf. It is possible that such a commitment will contain unbound variables representing the need to search for an agent suitable for delegation.
- *Joint commitments* (JC)—these commitments provides mutually linked commitments (of several agents) via decommitment rules. In a replanning situation the joint commitments proactively assure that the cost of the failure is minimized. An example of the use of a joint commitment is decommitting another agent's linked commitment as explained in the Section II
- *Minimal social commitment* (MSC)—is the classical type of decommitment, where the agent is required to notify the members of the team about its inability to achieve the commitment.
- *Relaxation* (R)—is a special decommitment, where the original commitment is replaced with a new commitment with relaxed condition and/or goal. The new commitment must be consistent with all other bound commitments. Provided that the bound commitment is of other agent, the relaxation must be negotiated. The asked agent tries to fit the requested relaxed commitment into its knowledge base and eventually use some other decommitment rules of other commitments to change it and fulfill the request.

During the replanning process, the preference relation over the commitments is TC $\succ$ IC $\succ$ D $\succ$ JC $\succ$ MSC. The preference of R can be arbitrary managed by the agent in consideration of current circumstances.

### D. Commitment Graph

Using the extended form of the social commitment we can propose a graph notation of the commitments. The mutual bindings and commitments form a commitment graph—Fig. 3. The commitment graph describes the same properties of the mutual decommitting as the logical notation.

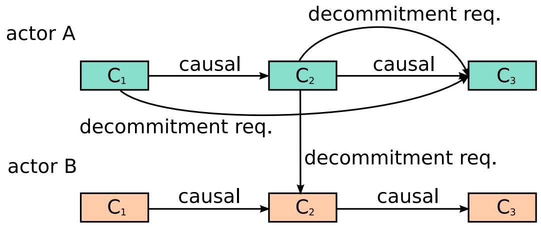


Fig. 3. Commitment graph—the *causal* links define the sequentiality of the commitments of each actor. The commitment $C_3$ of the actor A can be decommitted by both $C_1$ and $C_2$ commitments. The $C_2$ commitment of the actor B can be decommitted by actor A using the decommitment request.

The graph notation can be used to describe the process of the successive solving of the exceptional states. The process is based on the traversing through the commitment graph. The traversing starts with the first violated commitment. One of the decommitment rules is triggered (according to the violation type). As the decommitment rule is a commitment it starts an intetnion of the agent to terminate the commitment. In the case, that the intention is a decommitment request, the process crosses on the requested commitment (decommitment rule respectively) and starts one of the decommitment rules on the side of the requested commitment. Provided that the decommitment rule terminates the commitment wihout a need to request other decommitments, the process ends here and the violation is fixed.

### E. Deployment Scenario

The approach presented in this paper is being verified on a realistic simulation scenario—Fig. 4. The scenario is based on an island inspired by Pacifica Suite of Scenarios[2]. On the island, there are cities and a net of roads connecting them, but off-road movement is also enabled. There are also several seaports and airports. The scenario actors are several unit types (ground, armored, aerial or sea units), civilians and non-friendly units.

There are ground units, which are *Transporters* (can provide faster transportation of other unit(s), material or civilians), *Construction* (can repair damages or assemble/disassemble stationary units) and *Medical* (provides medical care for other units or some rescue operations). The *Armored* units for protection of other units or secure an area or convoy. The *Aerial*—the UAVs with an extended visibility range and *Sea* units for transportation over the water.

The scenario simulates limited information visibility and sharing. Due to this, the environment provides non-deterministic behavior from the single unit point of view. There are heterogenous independent self-interested units in the scenario that commit to the shared/joint goals. To fulfill the desired strategic goals in such environment, the units provide

[2] http://www.aiai.ed.ac.uk/oplan/pacifica

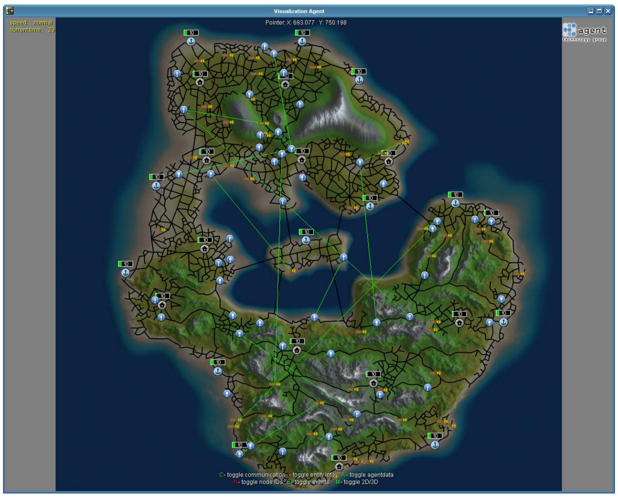


Fig. 4. Scenario island screenshot

complex cooperative actions on several levels of planning and control.

Planning and control of activities of individual units and actors in the scenario is loosely structured into three levels of detail. We recognize several layers of coordination and control:

- **Strategic layer:** The actors use aggregated meta-information from the tactical layer. This layer provides an overall strategic plan for middle and long term time horizon. High level planning and peer-to-peer coordination among the actors is possible (while non-transparent to the tactical level).
- **Tactical layer:** On this layer, the units use aggregated information from the individual layer, the information obtained through communication with each other and the information obtained from the strategic layer. The units and actors use classical planning and cooperation methods and can create new goals or adapt the goals received from the strategic level.
- **Individual layer:** On this layer, the units should perform reactive behavior based on obtained information and current goals.

The suggested coordination is hierarchical with respect of type of unit, area of operation and visibility. Three-layer architecture enables to separate middle- and long-term strategic planners from the real-time planning and control on the tactical and individual level. The strategic planner can utilize advanced planning methods with using aggregated meta-data from the whole system. On the other hand, the tactical planner has to provide real-time response and it uses limited information provided by individual layer of respective unit. On the tactical level local cooperation and information sharing of the field units is provided.

Each layer produces particular commitments and these commitments define the plan.

The strategic layer uses the HTN I-X planner [9] and a distributed resource allocation algorithm. The planner uses an abstract sub-domain derived from the scenario domain and produces an abstract plan. This plan is instantiated using negotiation about the resources—Fig. 5.

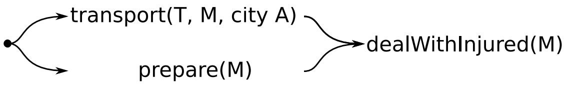


Fig. 5. Instantiated strategic plan—the medic unit $M$ was requested by the commander agent to fulfill a task: deal with the injured in city A, and it negotiated the transport with the transport unit $T$.

The instantiated plan is converted into commitments—Fig. 6. The conversion process creates a commitment according to the particular plan action ($\varphi = a$) and according to forward causality links of the plan.

The commitments of the tactical layer are based on strategic commitments. The layer uses negotiation to form the most suitable mutual commitments. The constraints for the negotiation respects the particular needs of the agents. The tactical commitments also define recommitments to the strategic layer and they can additionally refine some strategic commitment too. They are much more refined than the strategic commitment in the sense of spatio-temporal constraints, and particular world-states. The tactical commitments are most enriched by the $\lambda^*$ commitments. Thus, the most important part of the decommitting / replanning process is done by this layer.

An example of the tactical negotiation can be: A transport unit $T$ is planning the tactical commitment $moveto(l_1, l_2)$, it can find out it needs support from another unit. In this case, a negotiation process must find an appropriate support unit $S_p$ that proposes the most complying commitment (e.g. in terms of temporal constraints). If such a unit is found the JC is established, planned, and connected to other commitments in the knowledge base.

And finally, the individual layer plans commitments for later execution. These commitments copy the tactical commitments, but some of these can be omitted (e.g. $atPosition$ in the Fig. 6). Each individual commitment contains a decommitment request only to its parent commitment (from the tactical layer).

During the execution of the plan the commitments are processed. The commitment can evolve (Section II-C) according to the plan or due to unexpected environment interactions.

The monitoring of the commitments is triggered by a change of the world, e.g. a tick of the world timer, movement of a unit, a change of a world entity state, etc. The process evaluates all commitments in the actor's knowledge base. The value of the commitment defines the commitment state and can start the decommitting process.

One of the response to the unexpected situation can be relaxation. For instance, if a truck $T$ commits itself $C_1$ to move to position $l_1$ exactly at time $t_1$ and it faces an unknown risk combat zone the commitment has to be decommitted (because the time $t_1$ cannot be satisfied). So, $T$ tries to relax the commitment and thus changes the time constraint to $t_2$ (it plans a new route to $l_1$). And because the next bound commitment $C_2$ is constrained by time interval $\langle t_{min}, t_{max} \rangle$ where $t_2 > t_{min}$ and $t_2 < t_{max}$ the $C_2$ has not to be decommitted.

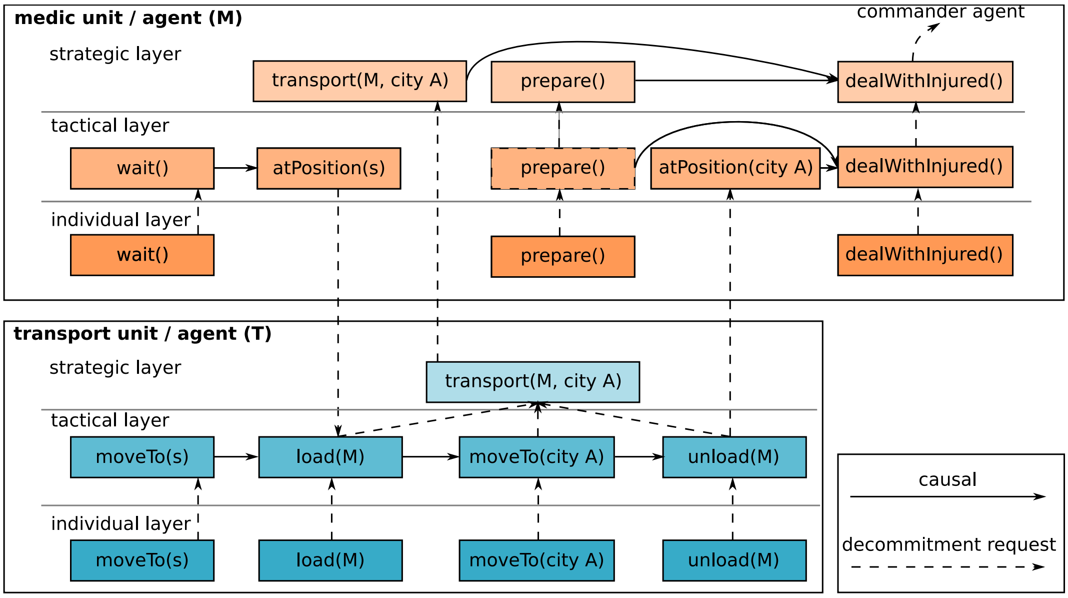


Fig. 6. Commitment bindings of multi-layer architecture for two units—the medic unit $M$ is committed to fulfill a task: deal with the injured in city A, and the transport unit $T$ is committed to transport the medic unit $M$ to city A. The figure shows the directions of the potential decommitment propagation among the layers of the actors.

Another example is the delegation of the commitment using negotiation. The current agent $A$ has to find a replacement agent $B$. If $B$ is found the agent's commitment passes on to the $B$. $B$ must integrate the commitment into its plan in the sense of the $\lambda^*$ commitments. The process of choosing $B$ is based on a measure of necessity to modify the current commitments of the proposing agent. For instance, let us have three trucks $T_1$, $T_2$ and $T_3$ and two builders $B_1$ and $B_2$. $T_1$ and $T_2$ commit $B_1$ and $B_2$ to move them to a location $l$. Let us assume that during the transport a problem occurs and as a result of it $T_1$ is no longer able to fulfill the commitment. In this situation, the commitment can be passed on either to $T_2$ or $T_3$. Since $T_3$ is idle, it is more appropriate to pass the commitment on to $T_3$ rather than $T_2$. $T_2$ would have to replan the current transportation commitment and all its successors.

The last example can be used to describe the usage of the nested commitments (commitment recurrence) too. In the case that $T_3$ cannot be accidently used, the $T_1$ cannot delegate the task to the $T_3$. This fact can be described using decommitting of the delegation decommitting rule in the $T_1$'s commitment base. Formally:

$$\begin{array}{l}(\text{Commit } T_1 \; \texttt{true} \; \texttt{is-transported}(B_1) \; \{ \\ \quad (\text{Commit } T_1 \; \texttt{immobile}(T_1) \; \texttt{delegated-to}(T_3) \; \{ \\ \quad\quad (\text{Commit } T_1 \; \texttt{immobile}(T_3) \; \texttt{true} \; \emptyset) \; \}, \\ \quad \ldots \textit{other decommitment rules} \\ \})). \end{array} \tag{10}$$

An agent can make a decision whether it is more suitable to re-run the strategic planning (which can be very costly and can lead to replanning of all plans of all other agents) or relax the commitment on its own (which would be probably a much less expensive operation).

## III. Related Work

Formalization of commitments has been extensively studied in the past using various formalisms, most of all building on and extending the BDI framework when describing obligations the agents adopt. Fasli [10] distinguishes two classes of obligations—general and relativized—and adoption of a social commitment by an agent is described as an adoption of a role. Thus, the agent promises its coherence with a (behavior) norm defined by the commitment. The framework extends BDI into a many-sorted first order modal logics to add concepts of obligations, roles and social commitments while it also uses branching temporal components from Computational Tree Logics (CTL) [11]. Besides strategies for adoption of social commitments by the agents the framework also defines strategies regarding conditions for a successful de-commitment from the agent's obligations.

Another formal representation of commitments considering temporal account has been introduced in [12]. CTL [11] has been extended to capture features not being usually considered in common approaches (but relevant for realistic environments), namely time intervals considered in commitments satisfaction, "maintenance" manner of commitments next to "achieve" manner of commitments and vague specification of time. Commitments have been formally defined using Backus-Naur Form as an $n$-tuple $(\text{Commit } id, x, y, p)$ where the commitment identified uniquely by its $id$ and the interpretation is that $x$ commits to $y$ to make the condition $p$ become true. The formal framework uses event calculus and defines operations $create(x, C)$, $cancel(x, C)$, $release(y, C)$, $assign(y, z, C)$, $delegate(x, z, C)$ and $discharge(x, C)$ above the commitments as well as new predicates $satisfied(C)$ and $breached(C)$ which evaluate the status of the commitments.

The past is considered linear while the future is branching. When created, the commitment is neither satisfied nor breached (the satisfaction of commitments is applied three-value logics). A commitment once satisfied or breached remains satisfied or breached once and for ever since the time.

Evolution of commitments in teamwork has been studied by Dunin-Keplicz [13]. Teamwork is explicitely represented using BDI framework by introducing a concept of a collective intention resulting in a plan-based collective commitment established within a group of agents adopting it. The teamwork consists of four consecutive stages—*(i) potential recognition*, *(ii) team formation*, *(iii) plan formation* and *(iv) team action*. The collective commitment based on a social plan (the collective intention) splits into sub-actions expressed as pairwise social commitments between agents. Establishment of the collective commitment consists in a consecutive execution of social actions defined at the particular stages: *(i)* `potential-recognition` → *(ii)* `team-formation` → *(iii)* `plan-generation` executed as `task-division` → `means-end-analysis` → `action-allocation` and *(iv)* `team-action` implemented as execution of respective actions allocated to each agent in the former stage. Naturally, the above-mentioned social actions are hierarchically bound from the first to the last stage. Dynamically evolving environment may cause unfeasibility of the allocated actions during the team action which results in a need for evolution of the collective commitment accordingly. In such a case, the maintenance of the collective commitment is achieved by invoking reconfiguration at the `action-allocation` level progressing upwards to higher levels of the hierarchy of social actions, possibly up to the `potential-recognition`. Finally, the collective commitment is adapted (another potential for the teamwork recognized) or dropped. The hierarchical manner of the reconfiguration allows for minimization of changes necessary to perform in order to adapt the collective commitment. The communication necessary for the reconfiguration is explicitly involved and formalized in the framework. Adaptation of the commitment is motivated by persistency of the joint intention which differs given a chosen intention strategy (*blind*, *single-minded* and *open-minded*). For the sake of not making the presented multi-modal logical framework even more complex and less tractable, temporal aspects of the cooperation are assumed to be expressed in a procedural way rather than by employing temporal and dynamic elements among the modalities used.

## IV. Conclusion and Future Work

This paper dealt with the problem of distributed planning used for replanning and plan repair processes. The classical work on commitments has been extended towards commitment recurrence for flexible and more expressive representation of replanning alternatives. Similarly, the termination condition has been defined as a specific type of commitment. The various types of commitments were classified according to the impact they may have on the other collaborating actors. This classification enables the agents to perform the right decision during the decommitment process.

This contribution represents only a starting point towards a more complex research effort that will be performed with social commitments within the context of distributed planning. We need to go beyond classification of the commitments to basic types and we need to design metrics and mechanisms that would allow agents to assign costs to each of the commitments. This will facilitate further research in design of scalable negotiation mechanisms allowing agents to negotiate the best commitments for their and social welfare perspectives.

Further integration of the HTN planning mechanism and the social commitments knowledge structure will be a critical research challenge we want to address. As described in the paper, we assume that the commitments resulted from the agent-oriented programming process and are uploaded from the agent's knowledge base. We plan to develop and design mechanisms for runtime creation of commitments from the hierarchical task networks, defining the planning problem and from the known hierarchy and knowledge of competency and abilities of the agents.

## References

[1] M. Wooldridge and N. Jennings, "Cooperative problem solving," *Journal of Logics and Computation*, vol. 9, no. 4, pp. 563–594, 1999.

[2] H. Levesque, P. Cohen, and J. Nunes, "On acting together," in *AAAI-90 Proceedings, Eighth National Conference on Artificial Intelligence*, vol. 2. Cambridge, MA, USA: MIT Press, 1990, pp. 94–99.

[3] E. H. Durfee, "Distributed problem solving and planning," in *A Modern Approach to Distributed Artificial Intelligence*, G. Weiß, Ed. San Francisco, CA: The MIT Press, 1999, ch. 3.

[4] M. E. DesJardins and M. J. Wolverton, "Coordinating a distributed planning system," *AI Magazine*, vol. 20, no. 4, pp. 45–53, 1999.

[5] E. Ephrati and J. S. Rosenschein, "A heuristic technique for multiagent planning," *Annals of Mathematics and Artificial Intelligence*, vol. 20, no. 1–4, pp. 13–67, 1997. [Online]. Available: http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1023/A:1018924209812

[6] M. M. de Weerdt, A. Bos, J. Tonino, and C. Witteveen, "A resource logic for multi-agent plan merging," *Annals of Mathematics and Artificial Intelligence, special issue on Computational Logic in Multi-Agent Systems*, vol. 37, no. 1–2, pp. 93–130, Jan. 2003.

[7] M. Wooldridge, *Reasoning about Rational Agents*, ser. Intelligent robotics and autonomous agents. The MIT Press, 2000.

[8] M. P. Singh, A. S. Rao, and M. P. Georgeff, *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA.: MIT Press, 1999, ch. Formal Methods in DAI: Logic Based Representation and Reasoning, pp. 201–258.

[9] A. Tate, "Intelligible ai planning," in *Research and Development in Intelligent Systems XVII (Proc. 20th ES)*, M. Bramer, A. Preece, and F. Coenen, Eds. Springer, 2000, pp. 3–16.

[10] M. Fasli, "On commitments, roles, and obligations," in *CEEMAS 2001*, ser. LNAI, B. Dunin-Keplicz and E. Nawarecki, Eds., vol. 2296. Springer-Verlag Berlin Heidelberg, 2002, pp. 93–102.

[11] E. Emerson and J. Srinivasan, "Branching time temporal logic," in *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, School/Workshop*, ser. LNAI, vol. 354. Springer-Verlag, 1988, pp. 123–172.

[12] A. Mallya, P. Yolum, and M. Singh, "Resolving commitments among autonomous agents," in *Advances in Agent Communication. International Workshop on Agent Communication Languages, ACL 2003*, ser. LNCS, F. Dignum, Ed., vol. 2922. Springer-Verlag, Berlin, Germany, 2003, pp. 166–82.

[13] B. Dunin-Keplicz and R. Verbrugge, "Evolution of collective commitment during teamwork," *Fundamenta Informaticae*, vol. 56, no. 4, pp. 563–592, August 2003.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Utility-based Model for Classifying Adversarial Behaviour in Multi-Agent Systems

Viliam Lisý, Michal Jakob, Jan Tožička, Michal Pěchouček
Gerstner Laboratory – Agent Technology Center
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{viliam.lisy,michal.jakob,jan.tozicka,michal.pechoucek}@agents.felk.cvut.cz

***Abstract*—Interactions and social relationships among agents are an important aspect of multi-agent systems. In this paper, we explore how such relationships and their relation to agent's objectives influence agent's decision-making. Building on the framework of stochastic games, we propose a classification scheme, based on a formally defined concept of *interaction stance*, for categorizing agent's behaviour as self-interested, altruistic, competitive, cooperative, or adversarial with respect to other agents in the system. We show how the scheme can be employed in defining behavioural norms, capturing social aspects of agent's behaviour and/or in representing social configurations of multi-agent systems.**

## I. Introduction

Recently, there has been a growing interest in studying complex systems, in which large numbers of agents pursue their goals while engaging in mutual interactions. Examples of such systems include real-world systems, such as diverse information and communication networks, as well as simulations of real-world systems, such as models of societies, economies and/or warfare. With the increasing diversity of these systems, there is a growing need to develop models which allow characterizing the behaviour of agents, and their interactions, in a compact form.

Fundamentally, the behaviour of an agent is driven by its objectives. In the absence of other constraints and influences, the agent is expected to perform actions which – incrementally but not necessarily monotonously – lead towards its objectives. In some situations, however, the way the agent acts can be affected by additional factors, be it the surrounding environment, agent's decision making capability or *its relations to other agents*. Knowing and understanding such factors may help in reasoning about the agent and in obtaining better prediction of its behaviour, compared to when only agent's overall objectives are considered.

Whereas the impact of the first two factors have been studied in several fields related to intelligent agents, including game theory and planning, comparatively less work seems to exist on relating agent's social relations, agents' objectives and the behaviour they ultimately execute. The aim of this paper is therefore to investigate and formalize *inter-agent relations* as an important behaviour-modifying factor in communities of social agents. Specifically, building on the framework of stochastic games and extending our earlier work [1], we propose a utility-based model which categorizes actions (and consequently also relationships among agents) as self-interested, altruistic, competitive, cooperative and adversarial. The concept of interaction stance allows to define, classify, and/or regulate agent behaviours not only with respect to agent's own objectives but also with respect to objectives of other agents in the system.

The rest of the paper is organized as follows. In Section II, we identify factors which affects how an autonomous agent pursues its objectives. Section III exposes the major part of the contribution – the classification model of agent's behaviour. Section IV shows several examples illustrating the application of the model. Section V overviews the related work and Section VI concludes with a summary.

## II. Agent's Decision Making

As already mentioned, the behaviour of an agent is primarily driven by its overall objectives (also referred to as *desires* in the BDI architecture [2]). Out of the factors which constraints/affect how the agent actually behaves, we can distinguish

- *social relations* to other agents – if an agent is cooperative with another agent, it may consider the other agent's objective in choosing the action it performs. If it has multiple alternatives how to fulfil its objectives, it can choose the one that would help the other agent or even follow a suboptimal course of action in order to help the other agent reaching its goals.
- *environment* can limit the available actions the agent is able to perform; additionally, the environment can make action outcomes non-deterministic due to its stochastic nature or interfering actions of other agents
- *decision-making capability* reflects the extent with which the agent is able to pursue its objectives. Some agents are purely reactive, others can use planning or predict changes of environment. The agent can be trying to reach a goal, but due to insufficient computational resources or some architectural limits end up choosing a suboptimal or even contra productive action. Decision making capability is closely related to the issue of *bounded rationality* [3].

Explicit consideration of different factors affecting agent's behaviour provides for a more detailed characterization of agent's decision making, in turn allowing to reason not only about agent's current behaviour but also about its behaviour in situations where some of these factors change.

For example, knowing than an agent A has a cooperative stance towards agent B allows to infer that, in the presence of agent B, the agent A will, if at all possible, perform actions that contribute towards achieving agent B's objectives if such actions are not necessarily optimum when agent A's objectives are considered alone (and which agent A would pursue if agent B was not present). Similarly, knowing that the environment prevents an agent from executing an action that contributes towards its objectives allows to infer that, should the environment state change favourably, the agent would execute the action (even if it has not performed it so far at all).

The rest of the paper is dedicated to formalizing the above given notions. Primarily, we focus on the effect of social relations, though the role of the environment is also considered to an extent.

## III. Interaction-based Classification Model

This section presents an interaction-based model developed for multi-agent systems with asymmetric agent's utility functions. The model allows classifying actions and their sequences with respect to their effects on utilities of agents in the system. It is based on the formalism of partially observable stochastic games [4] generalized to infinite state, action, and observation spaces and omitting the initial state and reward functions, which are substituted by agent utility functions.

The choice of utility functions as a linear combination of some predefined characteristics of the world as we define it later, over possibly more expressive reward functions is motivated primarily by the compactness of representation allowed by the former. The description of agents using utilities also seems closer to how people tend to think about agents – it is more straightforward to specify what an agent is trying to achieve in terms of the desired state of the world than trying to assign a reward for each possible state.

Although the underlying model of stochastic games is general enough to describe incompleteness of agent's knowledge about the world, we do not consider this factor in the current version of the model. Likewise, the explicit consideration of agent's possibly limited decision-making capability is currently not considered.

### A. Fundamental Definitions

**Definition 1.** The game model is a tuple $(\mathcal{I}, \mathcal{W}, \{\mathcal{A}_i\}, \{\mathcal{O}_i\}, \mathcal{P})$, where

- $\mathcal{I}$ is a finite set of agents (players) indexed by $1, \ldots, n$
- $\mathcal{W}$ is a possibly infinite set of all states of the world
- $\mathcal{A}_i$ is a set of actions available to agent $i \in \mathcal{I}$ and $\mathcal{A} = \mathsf{X}_{i \in I} \mathcal{A}_i$ is a set of joint actions where $a = \langle a_1, \ldots, a_n \rangle$ denotes a joint action
- $\mathcal{O}_i$ is a set of observations for agent $i$ and $\mathcal{O} = \mathsf{X}_{i \in I} \mathcal{O}_i$ is the set of joint observations where $o = \langle o_1, \ldots, o_n \rangle$ denotes a joint observation
- $\mathcal{P}$ is a set of Markovian state transition and observation probabilities, where $\mathcal{P}(w', o \,|\, w, a)$[1] denotes the probability that taking a joint action $a$ in a state $w$ results in a transition to state $w'$ and a joint observation $o$.

Another part of the formal model is the space of all possible world characteristics that could concern some agent. We refer to this space as the utility space and we assume that it is a subset of $\mathbb{R}^m$, where $m$ is the number of characteristics considered. A point in the utility space is a vector of world-specific values of these characteristics. Let us consider an example game in which an agent considers building a highway. Different characteristics the agent can consider in such a scenario include the impact on the traffic situation, financial cost or harm to the surrounding nature. Different agents have a different view on the importance of individual characteristics.

**Definition 2.** The *utility space* $\mathcal{U} \subseteq \mathbb{R}^m$ of a system is a vector space generated by all components of the utility functions in the system. We assume there exists a global utility function $\vec{u} : \mathcal{W} \to \mathcal{U}$ that assigns a utility vector to each state of the world.

Each agent in the world values the components of the utility space differently. A local scout organization cares more about trees and the local labour union values more employed workers. The government should consider both characteristics. The preferences of an agent over the utility space components (i.e. the characteristics) are expressed by a vector of real weights. The overall utility that an agent ascribes to a state of the world is then a preference-weighted sum of the utility components.

**Definition 3.** If $\vec{u}(w) \in \mathcal{U}$ is a point in the utility space corresponding to a state $w \in \mathcal{W}$ of a system and $\vec{u_A} \in \mathbb{R}^m, |\vec{u_A}| = 1$ is the preference vector of agent $A \in \mathcal{I}$, then the utility of the state $w$ for the agent $A$ is

$$u_A(\vec{u}(w)) = \sum_{i=1}^{m} u_A^i u^i = \vec{u_A} \cdot \vec{u}(w)$$

where the dot operation represents the dot product in $\mathbb{R}^m$. For a group of agents $\mathbf{G} \subseteq \mathcal{I}$ we define the preference vector as

$$\vec{u_{\mathbf{G}}} = \sum_{i \in \mathbf{G}} \vec{u_i}$$

Using this preference vector, the utility of the state of the world $w$ for the group is defined as

$$u_{\mathbf{G}}(\vec{u}(w)) = \vec{u_{\mathbf{G}}} \cdot \vec{u}(w)$$

[1] Since we do not focus on agents' observations, we will omit them in the following text, i.e. we will use this notation: $\mathcal{P}(w' \,|\, w, a)$.

### B. Normalization

Note that we require the preference vectors to be normalized for individual agents. The main reason for that is that we want to be able to compare and sum utilities of different agents. If we have two agents and both of them want to maximize only the amount of their money, we do not consider reasonable to say that one of them wants to maximize it more than the other. The difference comes only with introducing another utility component, e.g. harming innocent people. After that, the agents can differ in how much they want to maximize their money considering how their actions harm innocent people.

However, the preference vector for a group of agents is not normalized anymore. It expresses not only the preference relations between the different components, but also how big and consistent in the preferences the group is. A big group with agents that have random preference vectors has the group preference vector close to zero whereas a group of agents with identical preferences has a large preference vector in the direction of the preferences of individual agents.

The above definition of group preference vector also ensures the property of *associativity of subgroups*. If we have a set of agents grouped into several groups and we join the groups to create a bigger group including all the agents, the preference vector of the resulting group only depends on the individuals in the group and not on the subgroups they were previously part of. This property would not hold if we normalized group preference vector and combine the resulting normalized vectors.

### C. Action Utility

Although utility is usually defined for a state of the world, we also define it for an action in a state of world. The utility of an action is the difference between the utility of the state after the action is performed and the utility of the state before. However, such a definition does not take into account the potential non-determinism of action effects. Instead we define the expected utility of an action as the average utility of an action if it was performed an infinite number of times in the same state of the world.

**Definition 4.** If $w_0 \in \mathcal{W}$ and $a \in \mathcal{A}$ are a world state and a joint action, then we define the expected utility of the action $a$ in the world state $w_0$ as

$$\vec{eu}(a, w_0) = \left( \int_{\mathcal{W}} \mathcal{P}(w \mid w_0, a) \vec{u}(w) dw \right) - \vec{u}(w_0)$$

Note that $\vec{eu}(a, w_0) \in \mathcal{U}$.

### D. Taxonomy of Actions

There is nothing but joint actions in the real world. All agents are concurrently choosing from amongst a huge number of their individual actions and the world changes accordingly. Some actions can be easily attributed to a single agent, e.g. pressing a button, however many other actions may have multiple actors involved to a different degree, e.g. a car accident. The outcome of a joint action can also have multiple independent parts relevant to different agents.

Our main goal in this section is to classify joint actions as cooperative, self-interested, competitive and adversarial, with respect to different agents or groups thereof. Collectively, we refer to these classes as interaction stance. In order to do this, we need to separate the part of the action effect which is relevant for the group and to think about what could the group have done differently to change its influence to the part.

First of all, we need to get rid of the irrelevant components of the outcome. Consider two groups of agents with preference vectors $\vec{u_{\mathbf{G}}}$ and $\vec{u_{\mathbf{H}}}$. These two vectors generate a vector subspace of the utility space. If the directions of the preference vectors are the same or opposite, the subspace degenerates to a single dimension; otherwise it is a two-dimensional plane.

The model works also for one dimension, but let us assume a more general vector orientation. For any action, the important part for classification from the viewpoint of the groups $\mathbf{G}$ and $\mathbf{H}$ is *the projection of the action's expected utility vector to the subspace*. If we assume for a while, that a group of agents $\mathbf{G}$ is fully responsible for a joint action $a$ considering the groups $\mathbf{G}$ and $\mathbf{H}$, we can draw the action classification scheme as in Figure 1. Below we examine the different interaction stances in more detail. An action is considered self-interested from the group $\mathbf{G}$ if it increases the utility of the group.

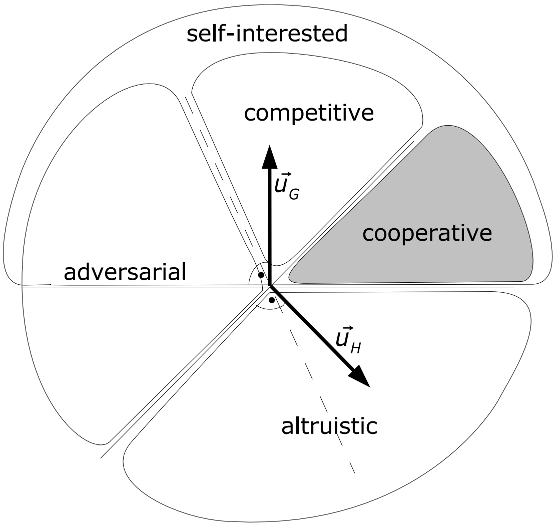


Fig. 1. Classification of a joint action.

**Definition 5.** We say that a joint action $a \in \mathcal{A}$ is *self-interested* for a group of agents $\mathbf{G} \subseteq \mathcal{I}$ in a state of the world $w \in \mathcal{W}$

$$\mathtt{si}_{\mathbf{G}}(a, w) \Leftrightarrow \vec{u_{\mathbf{G}}} \cdot \vec{eu}(a, w) > 0$$

We say that an action is cooperative with $\mathbf{H}$, if it changes the world in a way that increases the utility of both groups. Cooperative actions are symmetric. Actions that are cooperative from $\mathbf{G}$ towards $\mathbf{H}$ have exactly the same expected utility vectors as actions that are cooperative from $\mathbf{H}$ towards $\mathbf{G}$.

**Definition 6.** We say that a joint action $a \in \mathcal{A}$ is *cooperative* from a group of agents $\mathbf{G} \subseteq \mathcal{I}$ towards a group $\mathbf{H} \subseteq \mathcal{I}$ in a state of world $w \in \mathcal{W}$

$$\texttt{coop}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \Leftrightarrow \vec{u_{\mathbf{G}}} \cdot \vec{eu}(a, w) > 0 \;\&\; \vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w) > 0$$

An action is competitive, if it does not decrease the utility of the group $\mathbf{H}$ more than it increases the utility of the group $\mathbf{G}$.

**Definition 7.** We say that a joint action $a \in \mathcal{A}$ is *competitive* from a group of agents $\mathbf{G} \subseteq \mathcal{I}$ towards a group $\mathbf{H} \subseteq \mathcal{I}$ in a state of the world $w \in \mathcal{W}$

$$\texttt{comp}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \Leftrightarrow$$
$$\vec{u_{\mathbf{G}}} \cdot \vec{eu}(a, w) \geq |\vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w)| \;\&\; \vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w) < 0$$

An action is adversarial if it lowers the utility of the group $\mathbf{H}$ more then it increases the utility of the group $\mathbf{G}$ or even decreases the utilities of both the groups.

**Definition 8.** We say that a joint action $a \in \mathcal{A}$ is *adversarial* from a group of agents $\mathbf{G} \subseteq \mathcal{I}$ towards a group $\mathbf{H} \subseteq \mathcal{I}$ in a state of the world $w \in \mathcal{W}$

$$\texttt{adv}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \Leftrightarrow$$
$$\vec{u_{\mathbf{G}}} \cdot \vec{eu}(a, w) < -\vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w) \;\&\; \vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w) < 0$$

The last interaction stance not yet defined is the altruistic action. We define it as an action which helps some other agent while lowering the utility of the performing agent.

**Definition 9.** We say that a joint action $a \in \mathcal{A}$ is *altruistic* from a group of agents $\mathbf{G} \subseteq \mathcal{I}$ towards a group $\mathbf{H} \subseteq \mathcal{I}$ in a state of the world $w \in \mathcal{W}$

$$\texttt{alt}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \Leftrightarrow$$
$$\vec{u_{\mathbf{G}}} \cdot \vec{eu}(a, w) < 0 \;\&\; \vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w) > 0$$

At this point, we can define several classic game theoretic concepts in our framework. For example the utilities of a two player zero-sum game can be described by two vectors for which $\vec{u_A} = -\vec{u_B}$. Generally, we can define relationships between agents and group of agents based on the similarity of their weights of different utility components.

**Definition 10.** We say, that groups of agents $\mathbf{G}, \mathbf{H} \subseteq \mathcal{P}$ have *cooperative potential* if

$$\vec{u_{\mathbf{G}}}.\vec{u_{\mathbf{H}}} > 0$$

They have *adversarial potential*

$$\vec{u_{\mathbf{G}}}.\vec{u_{\mathbf{H}}} < 0$$

The definitions correspond to the correlation of agent payoffs used in game theory [5]. Two groups have cooperative potential if most of the self-interested actions of each group are cooperative with respect to the two groups. The adversarial potential occurs if most of self-interested actions are competitive or adversarial.

### E. Intentionality of Adversarial Action

The above defined concept of the adversarial action considers the effects of an action with respect to agents' individual and collective utilities. Classifying an agent as an adversary is based on the purely external analysis of agents behaviour, without taking into account agent decision-making capabilities and the influence of the environment on the executability and outcomes of its actions. The model presented so far can only represent whether the actions of a particular agent or a group of agents are helping or harming someone else.

In the real world, however, cooperative agents are often forced to choose the smallest evil, e.g. to choose an action that will harm the others least from all the available actions. In such situations, classifying the least harmful action as adversarial, as done by the model, might not be appropriate, as it may be the case that the agent has no other choice than to harm the other agents. A similar problem arises with the classification of the least beneficial action as cooperative if an agent can only perform beneficial actions.

This problem has been in part addressed in [6], where the concept of intentional adversarial action has been introduced. This definition assumes intentional adversariality based on agents' knowledge of adversarial nature of the particular action and agents knowledge of the existence of an alternative action that can be performed with less harmful effect. The definition could be loosely rewritten in the presented formalism as follows. If $\mathcal{A}^w_{\mathbf{G}}$ is the set of actions available to the group of players $\mathbf{G}$ in the state of world $w \in \mathcal{W}$ and $a_0 \in \mathcal{A}^w_{\mathcal{I}\setminus\mathbf{G}}$ is a combination of actions of the players that are not in $\mathbf{G}$, then an action $a = (a_0, a_{\mathbf{G}})$ is intentionally adversarial if

$$\texttt{adv}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \wedge \exists a' = (a_0, a'_{\mathbf{G}}) \in \mathcal{A} \text{ such that}$$
$$\vec{u_{\mathbf{H}}} \cdot \vec{eu}(a, w) + \vec{u_{\mathbf{G}}} \cdot \vec{eu}(a, w) <$$
$$\vec{u_{\mathbf{H}}} \cdot \vec{eu}(a', w) + \vec{u_{\mathbf{G}}} \cdot \vec{eu}(a', w)$$

This definition is quite strict. If a player with almost all its actions adversarial towards someone does not perform the single least harmful action, it is considered adversarial.

An alternative approach to the definition of an intentionally adversarial action and to effective separation of agent's intention and the interfering effect of the environment is based on the concept of *neutral behaviour*. Which action should an agent choose so that nobody could legitimately accuse it of acting self-interestedly or adversarially? The best solution in our opinion is to consider neutral an agent which chooses its action randomly with uniform distribution. With respect to this, the neutral outcome of an action is not the zero vector, but the average outcome of all the actions performable in a certain state of world. This corresponds to the centre of mass of all performable actions.

**Definition 11.** If $\mathcal{A}^w_{\mathbf{G}}$ is the set of actions available to the group of players $\mathbf{G}$ in the state of world $w \in \mathcal{W}$ and $a_0 \in \mathcal{A}^w_{\mathcal{I}\setminus\mathbf{G}}$ is a combination of actions of the players that are not in $\mathbf{G}$ then the *neutral utility* is

$$\vec{c} = \frac{1}{|\mathcal{A}^w_{\mathbf{G}}|} \sum_{a_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}}} \vec{eu}((a_0, a_{\mathbf{G}}), w)$$

If we want to classify an action, we first have to subtract the neutral utility from its expected utility and classify the difference as described above. Based on the neutral utility, an intentionally adversarial action would be defined as follows:

**Definition 12.** We say that a joint action $a = (a_0, a_{\mathbf{G}}) \in \mathcal{A}$ is *intentionally adversarial* from a group of agents $\mathbf{G} \subseteq \mathcal{I}$ towards a group $\mathbf{H} \subseteq \mathcal{I}$ in a state of the world $w \in \mathcal{W}$

$$\begin{gathered} \texttt{adv\_int}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \Leftrightarrow \\ \vec{u_{\mathbf{G}}} \cdot (\vec{eu}(a, w) - \vec{c}) < -\vec{u_{\mathbf{H}}} \cdot (\vec{eu}(a, w) - \vec{c}) \\ \wedge\, \vec{u_{\mathbf{H}}} \cdot (\vec{eu}(a, w) - \vec{c}) < 0 \end{gathered}$$

If the agent has no alternative to the action that is a subject of our investigation, its expected utility would be the same as the expected utility of the neutral action, and thus $(\vec{eu}(a, w) - \vec{c}) = 0$. This fact does not allow classifying the action as intentionally adversarial. The least harmful action from all possible actions also cannot be classified as intentionally adversarial. The last statement is generalized in the following proposition.

**Proposition 1.** *If there is an action $a = (a_0, a_{\mathbf{G}}); a_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}}$ intentionally adversarial from group $\mathbf{G}$ towards a group of agents $\mathbf{H}$ then there exists a sub-action $a'_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}}$ that forms an action that is less harmful to $\mathbf{H}$.*

$$\begin{gathered} \texttt{adv\_int}_{\mathbf{G}\rightarrow\mathbf{H}}(a, w) \Rightarrow\ \exists a'_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}} \\ \vec{u_{\mathbf{H}}} \cdot (\vec{eu}((a_0, {a'}_{\mathbf{G}}), w)) > \vec{u_{\mathbf{H}}} \cdot (\vec{eu}((a_0, a_{\mathbf{G}}), w)) \end{gathered}$$

*Proof:* Assume for contradiction that

$$\forall a'_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}}\ \vec{u_{\mathbf{H}}} \cdot (\vec{eu}((a_0, a'_{\mathbf{G}}), w)) \leq \vec{u_{\mathbf{H}}} \cdot (\vec{eu}((a_0, a_{\mathbf{G}}), w))$$

then using the definition of the neutral utility

$$\begin{aligned} \vec{u_{\mathbf{H}}} \cdot \vec{c} &= \frac{1}{|\mathcal{A}^w_{\mathbf{G}}|} \sum_{a'_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}}} \vec{u_{\mathbf{H}}} \cdot \vec{eu}((a_0, a'_{\mathbf{G}}), w) \\ &\leq \vec{u_{\mathbf{H}}} \cdot \vec{eu}((a_0, a_{\mathbf{G}}), w) \frac{1}{|\mathcal{A}^w_{\mathbf{G}}|} \sum_{a'_{\mathbf{G}} \in \mathcal{A}^w_{\mathbf{G}}} 1 \\ &= \vec{u_{\mathbf{H}}} \cdot \vec{eu}((a_0, a_{\mathbf{G}}), w) \end{aligned}$$

The inequality holds thanks to the assumption. Now look at the definition of intentional adversariality. The second condition in the definition is

$$\vec{u_{\mathbf{H}}} \cdot (\vec{eu}((a_0, a_{\mathbf{G}}), w) - \vec{c}) < 0$$

hence

$$\vec{u_{\mathbf{H}}} \cdot \vec{eu}((a_0, a_{\mathbf{G}}), w) < \vec{u_{\mathbf{H}}} \cdot \vec{c}$$

This contradicts the inequality above and thus concludes the proof. □ ■

The presented classification expects the agent to have full information about the world. This is usually not true, the information an agent has can be partial, incorrect, or the agent can even overestimate its capabilities. In this case the expected utility of an action changes to what agent believes is their expected utility and the perceived neutral utility becomes the average outcome of all the actions the agent believes it can perform.

### F. Traces of Actions

We can easily generalize the classification of actions to classification of runs in a multi-agent system in a straightforward way. The expected effect of the actions in a run on the utilities is the sum of the expected effects of the actions included in the run. Other definitions would be possible, but this one manifests the intentions of an agent with full information about the world, and is therefore most suitable for the classification of the interaction stance.

**Definition 13.** *Run* in a multi-agent system is a sequence of world states and actions

$$\rho = (w_1, a_1, w_2, a_2, ..., w_{k+1});\ w_i \in \mathcal{W}, a_i \in \mathcal{A}$$

where the state of the world is transformed from $w_i$ to $w_{i+1}$ via $a_i$.

**Definition 14.** Let $\rho = (w_1, a_1, w_2, a_2, ..., w_{k+1});\ w_i \in \mathcal{W}, a_i \in \mathcal{A}$ be a run of a multi-agent interaction, we define the *real utility* of the run

$$\vec{u}(\rho) = \vec{u}(w_k) - \vec{u}(w_1)$$

The *expected utility* of the run is

$$\vec{eu}(\rho) = \sum_{i=1}^{k} \vec{eu}(a_i, w_i)$$

The expected utility of a run can be classified in the same way as the expected utility of an action. By comparison of the of the utility vectors $\vec{u(\rho)}$ and $\vec{eu}(\rho)$, we can analyze how predictable or in a way intentional was the real outcome of a sequence of actions.

### G. Illustrative Example

To illustrate the use of the classification model introduced in the previous section, we apply it to a simulation of a flood relief operation. We consider three agents in the operation: the local government, humanitarian non-governmental organization (NGO) and separatists. The overall goal of the government, helped by the NGO is to stabilize the situation whereas the separatists want to take advantage of the situation is gain control over the affected regions.

Specifically, applying the proposed model, we identify the following utility components in the scenario: (1) the number of people with sufficient food and shelter, (2) the number of villages that are under the control of the government and (3) the state of the infrastructure damaged by the flood. One of the basic rules of the scenario is that the government cannot control a village, where people do not have enough food and shelter because of the riots that arise. The objectives of the three agents involved in the scenario are as follow. The government wants to maximize all three utility components. The NGO cares only about maximizing the number of people with food and shelter. Finally, the separatists also care about people's well-being, but prefer the government not to have control. Consequently, they do not want the infrastructure repaired, because it gives tactical advantage to the government.

If all three agents value all the utility components equally, then the normalized preference vectors, using the order in which the components were introduced above, are:

$$\vec{u}_{GOV} = (0.58, 0.58, 0.58)$$
$$\vec{u}_{NGO} = (1, 0, 0)$$
$$\vec{u}_{GNG} = (0.58, -0.58, -0.58)$$

Some of the relevant actions in the scenario are delivering food to a village, destroying food supplies in a village or rebuilding infrastructure in a village. Below, we classify these actions with respect to the government and separatist agent. If we project the expected utilities of the actions to the plane generated by their utility vectors, we get the situation depicted in Figure 2. A typical cooperative action of one of the players towards the other is delivering food to starving people. The expected utility of the action is $(1, 0, 0)$ and it improves the utility of both players to the same extent. A competitive action of the government towards the separatists is e.g. rebuilding infrastructure, with the expected utility of $(0, 0, 1)$. The utility the government gains from this action is the same as the loss of the separatists. An adversarial action of the separatists is destroying food supplies in a village with the expected utility of $(0, -1, 0)$. It lowers both players' utilities equally, because the food cannot be delivered to the starving people. Still, the separatists may choose to perform such an action with the intention to harm the government and with the anticipation of riots and subsequent government's loss of control over the village, which would eventually increase separatists utility.

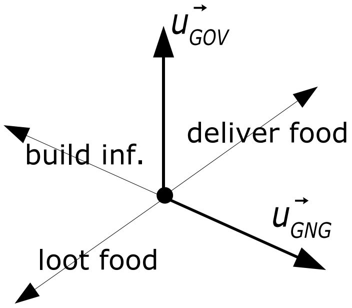


Fig. 2. Actions in the disaster relief scenario and their relation to the utilites of the Goverment and Separatists agent

## IV. Applications

In addition to providing formal grounding for a terminology used in categorising inter-agent relationships, the proposed scheme has several other applications.

### A. Interaction Norms

Exploring how expected and/or allowed behaviours of agents in open multi-agent systems can be described, monitored and possibly enforced is an active research topic. Most of the developed formalisms, however, focus on defining norms as restrictions on actions an agent must or must not take in particular situations, without explicit reference to utilities of other agents.

The concept of interaction stance, as formally introduced in this paper, allows to specify norms which involve such a reference. It can be e.g. stipulated that every agent must be cooperative with respect to the head agent of the community, or that it must not be adversarial to any other member of the community[2]. Instead of prescribing literally which behaviours are allowed, such *interaction-based norms* allows prescribing the behaviour *relative* to other agents. The advantage of interaction-based norms in their flexibility – should the utility of the administrator agent change (e.g. due resource congestion arising in the system), individual member agents must adjust their behaviour accordingly (e.g. stop bandwith-intensive transfers).

An important property of the proposed classification scheme is that it is operationalizable. Whenever an agent performs an action in the system, it is categorized and depending on the resulting category, respective norms can be applied.

### B. Agent Profiles

The interaction stance of an agent towards other agents (or classes thereof) can be made part of agent's profile. If combined with the knowledge of agent's base utility, a set of such profiles can provide for a compact representation of social relationships in agent community in way that allows reasoning about the behaviour of different (sub-)groups of agents in the system. Based on the profile of agents in a community and the profile of a potential new entrant, it can be determined whether the introduction of the new agent would benefit or harm the community.

### C. Case-based Reasoning

The compact description using agent profiles, each combining agent's base utility and its interaction stance towards other agents, can be further used in a case-based reasoning system. When an agent is to operate in a community with a particular configuration of agents, it can search the case base for a situation in which the same or similar agents with the same or similar interaction stances were involved. For example, a humanitarian relief operation can proceed differently if farmers are cooperating than if they are adversarial.

### D. Agent Design

Design of autonomous agents for open MAS is another application of the classification scheme. By taking into account the primary utilities of other agents in the system, the behaviour of the designed agent can be adjusted to implement

[2] Note that this is not possible for all combinations of agent's utlities.

the desired stance towards other agents in the system. Such an adjustment can be done off-line by a designer or on-line by the agent itself, provided agent's decision mechanism is flexible enough to realize such adjustment.

## V. Related Work

The different factors affecting how agents choose actions they perform have been widely studied in the literature. The impact of limited decision-making capabilities has been explored within the topic of bounded rationality (see e.g. [3]). The role of the environment in affecting agent's ability to achieve its objectives has been long-studied in planning; the concept of joint action capturing the effect of other agents' actions on the desired outcome of the action performed by the agent has been long-known in game theory and multi-agent reinforcement learning.

The concept of adversariality has been studied from game theoretical perspective with applicability in economical theories and wargaming [7]. The repeated games approach with incomplete information and knowledge was used to model attackers and defenders in information warfare [8]. In robotics domains (especially robocup soccer) the adversarial actors are preventing the other actors from effective achieving their goals [9], [10]. An incentive-based modelling and inference of attacker intent, objectives, and strategies has been reported in e.g. [11]. Recently behaviour of adversarial agents in multi-agent domains has been defined via motivation to cause a drop of agents' social welfare, even at the cost of the adversarial agents individual utility [1]. However, so far there seems to be no computer-science literature attempting to formally ground the otherwise frequently used informal notions of adversariality or altruism and relating them to agent objectives defined as utility functions.

## VI. Conclusion

In complex scenarios, agents do not always choose actions that lead optimally to the fulfilment of their objectives. The factors causing such a behaviour can be multiple, including agent's limited decision-making capability or the restrictive force of the environment. The paper focuses on the factor which has not yet been sufficiently addressed in this context – agent's social relationships to other agents. Agents may choose to perform actions not fully aligned with their objectives, if this helps or possibly prevents other agents from reaching *their* objectives.

In order to formalize such influence of social factors, we employ a formalism based on partially observable stochastic game to describe inter-agent interactions and define the concept of *interaction stance*, which categorizes agents' actions as self-interested, cooperative, competitive and adversarial. The classification is based on agents' preferences over different characteristics computed from the state of the system in consideration. We present two possible views of the classification. The first considers the effects of the actions of a group of agents towards other agents without taking into account actions available to the agents, whereas the other does take the real action options in account. The later approach allows differentiating between behaviours producing unintended negative effects and behaviours that are deliberatively adversarial. We show several applications of the model for the representation of norms, compact representation of social aspects of agent behaviour and representing social configurations of multi-agent systems.

In the future, we plan to include partial information about the world state, which can be represented via observations in the formalism, and extend the classification scheme to include agent beliefs about the world and the impact of their actions. Another important direction is the creation of methods for the reconstruction of agent utilities and their interaction stance from the observations of their actions in different social environments.

## Acknowledgements

Effort sponsored by the Air Force Office of Scientific Research, USAF, under grant number FA8655-07-1-3083. The U.S. Government is authorized to reproduce and distribute reprints for Government purpose notwithstanding any copyright notation thereon.

## References

[1] M. Pěchouček, J. Tožička, and M. Rehák, "Towards formal model of adversarial action in multi-agent systems," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM Press, 2006.

[2] M. E. Bratman, *Intentions, Plans, and Practical Reason*. Cambridge MA: Harvard University Press, 1987.

[3] A. Rubinstein, *Modeling Bounded Rationality*. The MIT Press, December 1997. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0262681005

[4] E. Hansen, D. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, 2004, pp. 709–715.

[5] V. Könönen, "Multiagent reinforcement learning in markov games: Asymmetric and symmetric approaches." Ph.D. dissertation, Helsinki University of Technology, 2004.

[6] M. Rehák, M. Pěchouček, and J. Tožička, "Adversarial behavior in multi-agent systems," in *Multi-Agent Systems and Applications IV*, ser. Lecture Notes in Computer Science, M. Pěchouček, P. Petta, and L. Z. Varga, Eds., vol. 3690. Springer, 2005, pp. 470–479.

[7] P. Lehner, R. Vane, and K. Laskey, "Merging AI and game theory in multiagent planning," in *Intelligent Control, Proceedings of 5th IEEE International Symposium on*, 1990., pp. 853–857.

[8] D. Burke, "Towards a game theory model of information warfare," Master's thesis, Graduate School of Engineering and Management, Airforce Institute of Technology, Air University,, 1999.

[9] R. M. Jensen, M. M. Veloso, and M. H. Bowling, "Obdd-based optimistic and strong cyclic adversarial planning." in *In Proc. of ECP01*, 2001.

[10] T. F. Bersano-Begey, P. G. Kenny, and E. H. Durfee, "Multi-agent teamwork, adaptive learning and adversarial planning in robocup using a PRS architecture," in *IJCAI97*, 1997.

[11] P. Liu and W. Zang, "Incentive-based modeling and inference of attacker intent, objectives, and strategies," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, USA: ACM Press, 2003, pp. 179–189.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# An Agent Based System for Distributed Information Management: a case study

Martijn Warnier, Reinier Timmer, Michel Oey, and Frances Brazier
Intelligent Interactive Distributed Systems, Faculty of Sciences
VU University Amsterdam, the Netherlands
{warnier, rjtimmer ,michel, frances}@cs.vu.nl

Anja Oskamp
Computer Law Institute, Faculty of Law
VU University Amsterdam, the Netherlands
a.oskamp@rechten.vu.nl

***Abstract*—Securely managing shared information in distributed environments across multiple organizations is a challenge. Distributed information management systems designed to this purpose, must be able to support individual organizations' information policies whilst ensuring global consistency and completeness of information. This paper describes a multi-agent based prototype implementation of a distributed information management system, for distributed digital criminal dossiers. The prototype implementation runs on the multi-agent platform AgentScape.**

## I. Introduction

MANAGING shared information securely and efficiently across multiple independent organizations is a challenge [2], [14]—the challenge this paper addresses. Individual organizations manage their own information locally on their own systems, according to their own information policies. If information is to be shared, however, additional system-wide policies and mechanisms are needed to manage global correctness and consistency.

One example of an environment in which information needs to be shared between multiple semi-independent organizations is the environment in which the Public Prosecution compiles and manages digital criminal dossiers. These criminal dossiers are currently being used in the Courts of Amsterdam and Rotterdam in a number of pilot studies. Earlier work in the Agent-based Criminal Court Electronic Support Systems (ACCESS) project [11], [12], [13] has focused on the high-level design and security of a distributed agent based architecture [4] that manages information based on the notion of *distributed digital criminal dossiers*. This paper describes a prototype implementation of a distributed information management system, that runs on the multi-agent platform AgentScape [7].

The main advantage of a distributed architecture is that it allows *physically* distributed information sources, such as Municipalities and the Prison Systems, to be, and thus remain, responsible for the integrity of their own information content in a digital dossier. By law, the Public Prosecution is responsible for the compilation and maintenance of criminal dossiers. Dutch law also dictates which information can and must be exchanged to facilitate compiling a criminal dossier[1].

In our approach the Public Prosecution has a central role and is responsible for providing the infrastructure that enables other organizations to securely add information and securely access information in criminal dossiers. Together, these organizations form a semi-open environment: an environment in which organizations are known and have control over their own information. This paper discusses some of the details involved in implementing a distributed agent-based information management system that allows *secure* information *exchange* across multiple organizations. In contrast, earlier work [1], [6] on distributed agent-based information management has mainly focused on *finding* information in *open* network, i.e., the Internet.

The remainder of this paper is organized as follows. The next section gives some background on the context in which the ACCESS prototype should function as well as on the AgentScape platform. Section III illustrates the design of the system in some detail and Section IV gives more information on the actual ACCESS prototype implementation. The paper ends with a discussion and conclusions.

## II. Background

This section briefly introduces the notion of *Distributed Digital Criminal Dossiers* compiled and managed by the Public Prosecution. See  [11], [12], [13] for more detail of compilation and management of distributed dossiers in complex and adaptive environments. This section also briefly introduces the AgentScape [7] agent platform on which the prototype runs.

### A. Context

In the semi-open environment in which the Public Prosecution compiles

and manages digital criminal dossiers, information is shared between multiple semi-independent organizations. On an experimental basis and in specific pilot studies such criminal dossiers have been used in the Courts of Amsterdam and

[1] Please note that all legal and procedural details discussed in this paper are interpreted in the context of Dutch law, but can be extended to legislation of other jurisdictions.

Rotterdam. This paper describes the architecture and implementation of a distributed multi-agent system architecture [4], designed to improve consistency, completeness, integrity and security of the information in digital criminal dossiers.

The physically distributed sources of information distributed over different organizations are depicted in Figure 1 below[2].

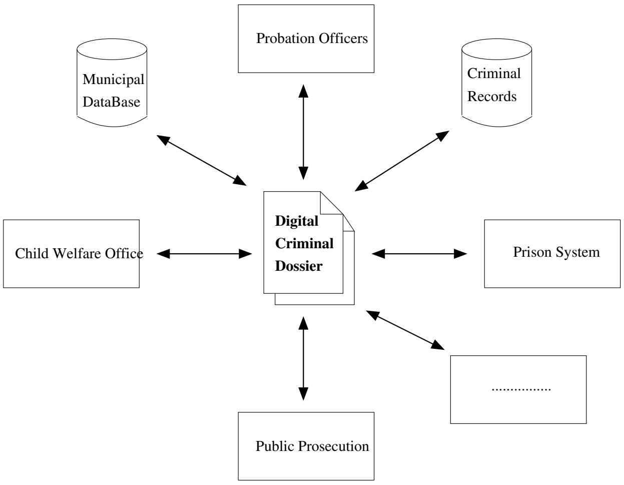


Fig. 1. Information Sources for a Digital Criminal Dossier

The Public Prosecution determines if (and when) an individual is to be prosecuted based on the law and the information available. Once decided, the Public Prosecution creates a criminal dossier for the case. Each newly created criminal dossier contains standard (meta-)data, and crime specific (meta-)data.

Different organizations are responsible for different information in the criminal dossier. Administrative information, for example, is managed and maintained by the defendant's local authorities[3] and information on a juvenile defendant's family situation and social environment is the responsibility of the Council for Child Welfare[4]. Distributing both the *data* (information) and the *responsibility* for the data ensures that information in digital dossiers is kept as up-to-date as possible. Changes in data are flagged by the relevant organizations and transmitted to the Public Prosecution for synchronization of the complete (distributed) criminal dossier.

The criminal dossier as a whole is the responsibility of the Public Prosecution whilst relevant data are maintained and stored by the organizations responsible for the data. The Public Prosecution is responsible for synchronization of the data within the digital criminal dossier as a whole and for the final version of the dossier that is sent to the judge for judgement in trial.

Note that the term 'dossier' is overloaded. In this paper the focus is mostly on *criminal* dossiers. Other dossiers include administrative dossiers, dossiers managed by the Council for Child Welfare, and, in general, all documents types that can contain references to other documents. In the remainder of this paper the term 'dossier' refers to dossiers in general, while 'criminal dossiers', 'administrative dossiers' etc, refer to specific dossier types.

### B. AgentScape

AgentScape [7] is a multi-language mobile agent middleware platform designed to support scalable, secure, distributed multi-agent applications. Agents can migrate between virtual domains called *locations* and can negotiate resource requirements before migrating [5]. An AgentScape location consists of one or more hosts running the AgentScape middleware, typically within a single administrative domain. Each AgentScape location runs a *location manager* process on one of its hosts. Each host has its own *host manager*. An example of two AgentScape hosts is depicted below in Figure 2. Multiple locations together with a lookup service form an AgentScape *world*: a distributed multi-agent system. The prototype implementation, discussed in this paper, is an example of an AgentScape world.

The AgentScape *kernel* provides low-level secure communication between the higher level middleware processes and between agents, and provides secure agent mobility. *Agent servers* provide language dependent run-time environments for agents, *Web service gateways* [8] support Web service access and monitoring using the SOAP/XML protocol. Two versions of the kernel have been implemented, in Java and in C.

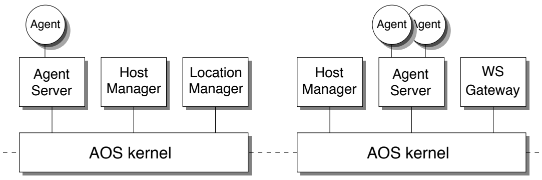


Fig. 2. The AgentScape middleware architecture.

## III. Application Design

The ACCESS prototype application is designed to be managed decentrally. Data is stored in digital criminal dossiers, fields of which are distributed among different locations (organizations). The contents of each field in a criminal dossier is automatically updated by the location responsible. This section describes how the information and locations are structured.

### A. Distributed Digital Criminal Dossiers

Dossiers are implemented with a light weight 'skeleton' framework based on XML. Fields in this XML document

[2] In this context, the Police have a special role. The Police provide information needed for compiling criminal dossiers by others (Public Prosecution, Probation officers etc), but cannot, for example, change information in criminal dossiers once a criminal dossier has been created by the Public Prosecution. Thus, information exchange between the Police and other organizations does not occur via distributed digital criminal dossiers.

[3] In the Dutch context all citizens are obliged to be registered in one and only one Municipality. Municipalities are responsible for keeping administrative records of their residents. Please note that not all countries have such administrative records at the governmental level, e.g. the USA do not maintain any administrative records at this level.

[4] In the Dutch context, it is the task of the Council for Child Welfare to investigate all aspects of a crime that involves minors, including the family situation and other relevant social factors, and to provide a motivated advice for suitable punishment (if applicable) of the suspect to the Courts.

contain keywords and references to other (possibly, but not necessarily, XML-based) documents. The distributed nature of the dossier is acquired through references to documents on both local and remote systems.

The root of each criminal dossier, created by the Public Prosecution, always includes the dossier number, creation date, access control lists, type of offense, (a reference to) the defendant's administrative data (in an administrative dossier), and the original police report concerning an incident. It also includes mandatory and optional information for each specific offence. Such information is used by the Public Prosecution to check completeness of a criminal dossier—a criminal dossier is only complete if all mandatory information is present. It provides the structure needed for automated completeness checks [12].

This paper assumes that standard XML templates exist for each type of offence, and that these templates are used by the Public Prosecution to structure the meta-data in a criminal dossier[5].

A distributed digital criminal dossier thus consists of a number of documents in a networked structure distributed over physically distributed locations with the Public Prosecution as the central coordinator. Note that a digital criminal dossier may only be referenced by other digital criminal dossiers. All digital criminal dossiers themselves are controlled by the Public Prosecution, an important security requirement. The root of each criminal dossier, maintained by the Public Prosecution, also contains information such as logging information on who altered or accessed information when, and when the last backup of the dossier was made.

### B. Principals

Many users interact with the information management system (using agents) concurrently, from different locations (organizations). These users can be divided into three types with different administrative rights:

1) *Users of the information management system*
Users of the information management system interact (read, write) with dossiers to perform their jobs as judges, (legal) clerks, public prosecutors, etc. As users of the information management system they cannot add or remove dossiers and templates, and they don't have permission to change the configurations of the system.
2) *System administrators with administrative rights of the infrastructure*
Administrators can be further divided into world and location administrators. The world administrator, a person assigned this role by the Public Prosecution, maintains the AgentScape world. Note that the world includes the lookup service and the central user and template repositories (also see Section IV-C). Location administrators, are assigned their roles by the locations they represent, have the task to maintain local configurations and infrastructure, i.e., the host machines, within their location. Both types of administrators can add users to the system and specify their roles (judge, prosecutor etc).
3) *The owner of the criminal dossiers—the Public Prosecutor*
The Public Prosecutor is the owner of all criminal dossiers. The Public Prosecutor is the only person who can create (or destroy) new criminal dossiers. The Public Prosecutor assigns access rights to roles (see below), and roles to users.

Note that in our approach, access to (and subsequent manipulation of) a dossier by users of the first type, users of the information management system, is only possible via agents. The owner of the criminal dossiers, the Public Prosecutor, also uses agents to create and delete dossiers. These agents are their only means of accessing the system—the 'single point of entry' for each user into the information management system.

### C. Access Control

The semi-open nature of the application domain makes access control a particularly interesting challenge.

The solution proposed in this paper is based on a two-tier access model. The first level in the model is based on role based access control (RBAC) [10]. Each 'role', such as judge, lawyer, public prosecutor, or clerk, has certain rights with regard to the criminal dossier. This typically depends on specific security policies, for example, a clerk may only add information to a criminal dossier, not delete or modify anything, while a public prosecutor may change existing information in the criminal dossier and even create new criminal dossiers. Roles are also distinguished for all of the organizations associated with a dossier. The role based access control list is generated from the high level security policies. Each template includes the RBAC list associated with the offence for which the template has been designed, specifying which access rights are associated with which role.

Additionally, the second access level uses access control lists (ACL) [3] to determine the access rights of an individual user, within each of these organizations, *after* the RBAC list has granted access based on the user's role. Each individual dossier, i.e., an instantiation of a template, specifies, in its meta data, the names of specific users and their access rights: it specifies per user who may read or change, information in a dossier. For example, suppose that a specific RBAC assigns judges (those with the role of judge) read access to a specific type of dossier. The ACL, however, specifies which individual judges have permission to read a specific instance of a dossier. It may specify that only Judge A has read access. In this case, Judge B, although a judge, may not read the contents of this particular dossier instance. Thus in order to read fields in a digital dossier (via an agent) a user not only has to be assigned a specific role, he/she also has to be on the access control list of a specific dossier.

This distinction between roles and individual users is crucial in all dynamic environments. Security policies are based on this distinction. Security policies based on *roles* are relatively

[5] See [12] for a proposal to use automatic clustering techniques to construct (part of) this information automatically.

static (or at least ‘long lived’) and are typically globally valid (at all possible locations), while *individual*-based access control lists are typically dynamic (or ‘short lived’) and often apply to specific dossiers. Individual-based control lists change, for example, when a dossier is handed over to another clerk within the same organization, or to another judge. The combination of static and dynamic access control rules are designed to limit ‘label creep’ [9], (i.e., that access control regulations make *all* data more difficult to access, including the less sensitive information.) Section III-D describes how this two-tier access control mechanism is represented in dossier templates and instances. Note that the ACL is not mandatory—if not specified the globally defined RBAC lists are used to regulate access to dossiers.

Once roles have been distinguished and access rights assigned to users, users need to be able to be identified by the different systems within the different organizations. Authentication is needed. In practice, authentication is done by assigning each user a name and a password. Each username is assigned one or more roles. This information is stored in a central user data repository[6] which is unique for the specific environment, i.e., an ACCESS world and managed by the world administrator. This repository can only be accessed by users within this world who have been assigned the role and the right to maintain this information.

Users, however, do not interact directly with any one of the systems or repositories. A user interacts with an agent that represents that user, see Section IV-A below. Once authenticated, a user’s agent is provided with a unique set of credentials with which his/her agent can be identified/authenticated.

Note that access control is not the only security issue in this environment. Other security features, such as secure communication between location, backup mechanisms, integrity checks and others, are discussed in detail in [13].

### D. Data Types

As stated earlier, dossiers are structured according to templates. Which template is used depends on the offence (for criminal dossiers) and may also depend on other information (e.g. administrative dossiers, probation dossiers, etc). A template specifies the constraints and the (type of) information that a dossier must or may contain. The listing in Figure 3 shows a (simplified) example of a template in XML format. The template describes administrative dossiers that store information of a subject.

This template shows a role based access control list (RBAC). The RBAC list in a template states the permissions each of the roles have on dossiers that are based on this template. The RBAC list in the template in Figure 3 states that (all) mayors have the right to both read and write entries in dossier instances (but not in dossier templates) and that all mayors furthermore have the right to change the (optional) ACL of a dossier instance. Administrative clerks may both read and write administrative dossiers based on this template and judges are only granted read access.

```
<Template>
<Meta>
<Name>AdminInfo</Name>
<ID type="Value" mandatory="true"
    content="Integer" />
<RBAC>
    Mayor:R-W-ACL,
    AdminClerk:R-W,
    Judge:R
</RBAC>
<ACL type="Value" mandatory="false"
     content="String List"/>
</Meta>
<Fields>
<Field name="Name" mandatory="true"
     type="Value" content="String"/>
<Field name="Title" mandatory="false"
     type="Value" content="String"/>
<Field name="SocialNum" mandatory=true"
     type="link" content="SocNum"/>
</Fields>
</Template>
```

Fig. 3. An example of a dossier template containing administrative information

The template also names the fields in a dossier instance, including their format, and specifies whether a field is mandatory or optional. In this listing, “Name” and “SocialNum” (social security number) are mandatory, “Title” is optional. Name and Title are String values, SocialNum is a link to a dossier. The link must in this case, always point to a dossier of type “SocNum”.

Depending on their assigned roles and permissions, users may be able to modify a dossier instance, however they are never allowed to modify the template. The RBAC list inside a dossier can only be modified by the world administrator, if permission is given by a public prosecutor. Note that the RBAC list is mostly static and will typically not change often.

The following listing shows an example of an instance of part of the administrative dossier, based on the template “AdminInfo” depicted in Figure 3. Consistency of required fields and types are checked automatically. More sophisticated consistency checks (spanning multiple dossiers) are performed by specialized agents, see Section IV-A. As required by the template, string values for the mandatory fields “Name” and “SocialNum” are specified.

The mandatory social security number field “SocialNum” contains a link (format `[refId]@[location]`) pointing to dossier ID 12432, located at location “SocNumRepos”. The link points to a dossier possibly at another location. The precise content of this dossier is beyond the scope of this paper. In the AgentScape implementation of the ACCESS prototype, different locations store different kinds of information. Links point to different types of documents, e.g. pdf files (such as the original arrest warrant), movie files, dna profiles and other document types, stored locally or remotely.

The administrative dossier has the RBAC list inherited from the administrative dossier template. Note that the RBAC

[6] A data repository is typically a database running on a location.

```
<Dossier>
<Meta>
<Template>AdminInfo</Template>
<ID value="123876"/>
<RBAC> <!--Inherited from template-->
     Mayor:R-W-ACL,
     AdminClerk:R-W,
     Judge:R
</RBAC>
<ACL> Judge:Judy:R <!-- only Judge Judy
   is allowed to read this dossier.
   Access rights for mayors and
   administrative clerks are still
   determined by the RBAC list. -->
</ACL>
</Meta>
<Fields>
<Field name="Name" value="George"/>
<Field name="Title" value="Dr"/>
<Field name="SocialNum"
      value="12432@SocNumRepos"/>
</Fields>
</Dossier>
```

Fig. 4. An example of an (administrative) dossier instance –based on the template in Figure 3– containing administrative information

inside a dossier instance is a local cache of the template RBAC and is only used in case of malfunctions (e.g. when the network is down). But whereas the RBAC specifies the permissions for all dossiers based on this template, the dossier ACL specifies the permissions for this specific dossier, thereby refining the RBAC list. For the dossier instance in Figure 4 the ACL refines the access of judges: According to the RBAC list, of the template in Figure 3, all judges are allowed to read administrative dossiers of this type. The ACL of the administrative dossier instance refines the RBAC and restricts the read access to only one judge,the judge named “Judy”. The ACL does not specify any information on users with role “Mayor” or “AdminClerk”. Thus users that have these roles are not further restrained in their access rights. In summary:

- The dossier ACL is optional. If the ACL is not specified in the dossier, then the RBAC list in a document’s template determines the access rights assigned to users on the basis of the roles they have been assigned.
- The dossier ACL is more restrictive than the template RBAC. If a user’s role is not assigned specific rights in a document template’s RBAC list, the user cannot be assigned these rights in the dossier’s ACL. If this error occurs, it is noted, and access denied.
- If the dossier ACL contains a rule for role R, then users with role R will only be granted access if he/she is mentioned in this rule. (If the dossier ACL does not contain a rule for role R, then users with role R automatically gain all rights from the template’s RBAC list.)

Note once more that a dossier ACL is a refinement of a template’s RBAC list. A dossier ACL can be made whenever appropriate.

### E. Data Repositories

Within the ACCESS prototype application, information is stored at different locations. There is a central repository for user information and templates (managed by the world administrator), and one or more central repository for the root files of individual distributed digital criminal dossiers. These criminal dossiers contain references to information stored decentrally in repositories managed by repository owners responsible for the contents.

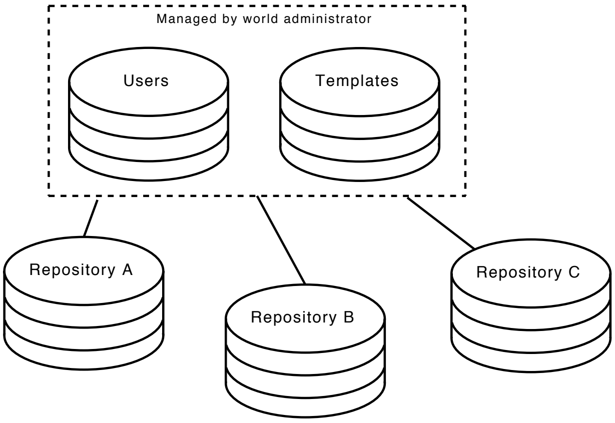


Fig. 5. ACCESS world organization

Users of the information management system at one location can include references to dossiers at other locations in their own dossiers, allowing information at the different locations to be managed by independent owners. Consequently, all updates to a dossier X at location A will automatically be part of all dossiers referring to dossier X from all other locations.

The ACCESS prototype is implemented as an AgentScape world. The world has its own lookup service, with the names of locations and their addresses. Access to services or repositories, outside the AgentScape world is only possible via a WS-Gateway [8]. This issue is not further addressed in this paper. The complete ACCESS world is illustrated in Figure 5.

## IV. Prototype Implementation

The ACCESS prototype is agent-based and runs on AgentScape. The implementation is not AgentScape specific and can, with some minor changes, be implemented in other agent platforms. The prototype supports communication between the agents and various repositories, and implements the security policies described above. The architecture of the ACCESS prototype is described below.

### A. Agents

The ACCESS architecture distinguishes a number of different types of agents, of which the first two form an intrinsic part of the system.

1) *Repository agents*
Repository agents regulate access to documents in repositories. Each repository agent is responsible for the documents in one repository. Repository agents regulate access to documents on the basis of agent credentials,

the policies specified in the RBAC lists of document templates, and the ACL of specific dossiers. Note that each repository agent has access to the templates and user information to check if a particular user agent (actually, the user behind the agent) is allowed to perform certain operations.

2) *User agents*
User agents represent the users of the information management systems. These agents interact with the repository agents to acquire access to a repository, i.e., each individual user agent sends the credentials of the user they represent, their dossier ID and the actions the agent wants to perform to the relevant repository agent. Once authorized, a user agent can then perform the action requested e.g. read, write (part of) the dossier managed by the repository agent.

3) *Functional agents*
Functional agents add new functionality to the ACCESS prototype. They can be inserted at runtime. Typical examples include: (i) *Consistency agents* are responsible for consistency checks across different repositories, between different dossiers. The consistency agent at the Public Prosecution is responsible for consistency of the information across repositories, as specified in the crime related template. (ii) *Completeness agents* are responsible for the completeness of the information in criminal dossiers. (iii) *Timeliness guarding agents* monitor time constraints (e.g. statute of limitations) with and across dossiers and take action when necessary.
Other dedicated agents can be deployed for specific tasks, guarding other specific types of constraints and regulations.

Depending on the specific requirements of a domain and the underlying infrastructure, different choices can be made with respect to the agents deployed. For example, repository agents may be responsible for flagging constraint violations (e.g. age $< 18$) or dedicated agents may be deployed for this purpose. The ACCESS prototype currently supports the first two types of agents and examples of agents of the other types.

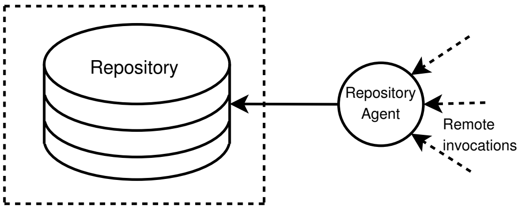


Fig. 6. Repository access by agent

AgentScape is responsible for running the agents and for the communication between agents. An agent's request (for access to a dossier at a different location, in fact, a remote method invocation) is always forwarded to the relevant repository agent, possibly at a different location, on a different host. Figure 6 shows how a repository agent provides other agents access to a repository.

One of the main features of the ACCESS prototype is that it supports incremental design. Agents can be added to the system as needed, adding new functionality, in fact, at runtime.

## B. Distribution

One of the main concepts in the ACCESS prototype is that each individual organization manages its own dossiers in its own repositories locally, within an AgentScape location. These locations are (often physically) distributed. A lookup service provides the names and addresses of the repositories within a world. This lookup service contains all the information for a single ACCESS application, i.e., for one Public Prosecution office[7].

## C. System Architecture

To access dossiers in the ACCESS world, a user of the information management system interacts with his/her user agent (an AgentScape agent). All communication with other agents is performed through this agent. The user's agent uses the user's credentials (derived from username and password), to acquire access to the different repositories.

In the current implementation, the user is presented with a graphical user interface (GUI) that communicates with the data repositories through the user agent. While the actual GUI window is running outside of AgentScape, all communication and authentication checks run inside the AgentScape platform in the ACCESS world. For an overview see Figure 7.

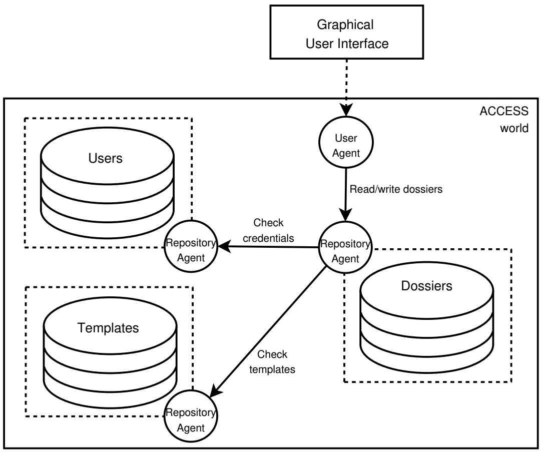


Fig. 7. Architecture of the ACCESS prototype

In theory, all users of the information management system could use their user agent to gain access to all dossiers in all repositories. In practice, however, whether they are actually allowed to view or edit the dossiers, depends on the agent's credentials, the dossier template's RBAC list and the dossier's ACL.

[7] When creating a world, the lookup service is the first thing to be started. Then at the central location, i.e., the Public Prosecution, the user and template repositories are created. These repositories are managed by the world administrator: the user that created the AgentScape world.

## V. Discussion & Conclusions

Information management in a distributed environment, such as the Public Prosecution and the organizations with which it cooperates, is an inherently difficult task. The use of multi-agent technology has clear advantages, as this paper demonstrates.

The basic system provides distributed and secure access to criminal dossiers for all users inside the ACCESS prototype. All information in the distributed digital criminal dossiers is maintained by local organizations, such as Municipalities, the Public Prosecution and others, providing local control and global access to information. In addition, information is always kept as up-to-date as possible, e.g. if a defendant moves, this is immediately reflected in the criminal dossier, once a Municipal updates this information in the (linked) administrative dossier.

Additional functionality can be added gradually in a modular fashion by means of new dedicated agents. Local organizations can add their own specialized agents and more globally useful agents can be distributed to all organizations. Again allowing maximum local control and global cooperation.

From a security point of view, as long as the computer system of the Public Prosecution has not been compromised, the digital criminal dossiers are not at risk. The computer systems of the Public Prosecution, however, need to be trusted completely. The lookup service, dossier templates and global authentication mechanism are all hosted by the Public Prosecution. If other systems are compromised the digital criminal dossiers are not necessarily effected. Automatic consistency checking, performed each time part of a dossier is altered, detects modifications. If these modifications are unwarranted and detected by both the dedicated agent and the user, once informed, the original data can be restored from a secure backup service.

The current prototype implementation in AgentScape supports several hundred dossier instances distributed over half a dozen locations. Future research will examine how well the implementation scales in a large scale distributed environment (10,000+ dossier instances on 50+ hosts).

The complete ACCESS prototype including a complete distribution of AgentScape can be found at http://www.agentscape.org.

## Acknowledgments

This research is supported by the NLnet Foundation, http://www.nlnet.nl, and is conducted as part of the Agent-based Criminal Court Electronic Support Systems (ACCESS) project, http://www.iids.org/access, initiated by the VU University Amsterdam together with the Courts of Amsterdam and Rotterdam, and financed by the Dutch Court for Jurisdiction (Dutch: Raad voor de Rechtspraak) and by the NWO TOKEN program.

## References

[1] J. Dale. *A Mobile Agent Architecture for Distributed Information Management*. PhD thesis, University of Southampton, 1997.

[2] C. A. Ellis and G. J. Nutt. Office Information Systems and Computer Science. *ACM Comput. Surv.*, 12(1):27–60, 1980.

[3] C. Kaufman, R. Perlman, and M. Speciner. *Network Security, PRIVATE Communication in a PUBLIC World*. Prentice Hall, 2nd edition, 2002.

[4] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink, 2003.

[5] D. G. A. Mobach. *Agent-Based Mediated Service Negotiation*. PhD thesis, Computer Science Department, Vrije Universiteit Amsterdam, May 2007.

[6] L. Moreau, N. Gibbins, D. DeRoure, S. El-Beltagy, W. Hall, G. Hughes, D. Joyce, S. Kim, D. Michaelides, D. Millard, et al. SoFAR with DIM Agents: An Agent Framework for Distributed Information Management. In *Proceedings of The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents*, pages 369–388, 2000.

[7] B. J. Overeinder and F. M. T. Brazier. Scalable middleware environment for agent-based internet applications. In *Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, volume 3732 of *Lecture Notes in Computer Science*, pages 675–679, Copenhagen, Denmark, June 2004. Springer.

[8] B. J. Overeinder, P. D. Verkaik, and F. M. T. Brazier. Web service access management for integration with agent systems. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC)*. ACM, March 2008.

[9] A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. *IEEE Journal on selected areas in communications*, 21(1), 2003.

[10] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *Computer*, 29(2):38–47, 1996.

[11] M. Warnier, F. M. T. Brazier, M. Apistola, and A. Oskamp. Distributed Digital Data: Keeping files consistent, timely and small. In *Proceedings of the eGovernment Interoperability Campus 2007 Conference (eGov-INTEROP'07)*, 2007. to appear.

[12] M. Warnier, F. M. T. Brazier, M. Apistola, and A. Oskamp. Towards automatic identification of completeness and consistency in digital dossiers. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Law (ICAIL'07)*, pages 177–182. ACM Press, 2007.

[13] M. Warnier, F. M. T. Brazier, and A. Oskamp. Security of distributed digital criminal dossiers. *Journal of Software*, 3(3):21–29, mar 2008.

[14] P. Yalagandula and M. Dahlin. A scalable distributed information management system. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 379–390. ACM Press New York, NY, USA, 2004.

# 3^rd^ International Symposium Advances in Artificial Intelligence and Applications

DEDICATED TO THE MEMORY OF PROFESSOR RYSZARD S. MICHALSKI

INTERNATIONAL Symposium on Advances in Artificial Intelligence and Application (AAIA) will be held within the framework of the International Multiconference on Computer Science and Information Technology, and will be co-located with the XXIV Fall Meeting of Polish Information Processing Society. Each year during the AAIA we plan to celebrate one outstanding Polish scientist who contributed to the field of Artificial Intelligence.

AAIA'08 will bring researchers, developers, practitioners, and users to present their latest research, results, and ideas in all areas of artificial intelligence. We hope that theory and successful applications presented at the AAIA'08 will be of interest to researchers and practitioners who want to know about both theoretical advances and latest applied developments in Artificial Intelligence. As such AAIA'08 will provide a forum for the exchange of ideas between theoreticians and practitioners to address the important issues.

Papers related to theories, methodologies, and applications in science and technology in this theme are especially solicited.Topics covering industrial issues/applications and academic research are included, but not limited to:

- Knowledge management
- Decision Support System
- Approximate Reasoning
- Fuzzy modeling and control
- Data Mining
- Web Mining
- Machine learning
- Neural Networks
- Evolutionary Computation
- Artificial Immune Systems
- Ant Systems in Applications
- Natural Language processing
- Image processing and understanding (interpretation)
- Applications in Bioinformatics
- Hybrid Intelligent Systems
- Granular Computing
- Architectures of intelligent systems
- Robotics
- Real-world applications of Intelligent Systems

INTERNATIONAL PROGRAMME COMMITTEE

**Ajith Abraham,** Norwegian University of Science and Technology, Norway

**Janez Brest,** University of Maribor, Slovenia

**Wlodzislaw Duch**, Nicolaus Copernicus University, Poland

**Krzysztof Goczyla,** Gdansk University of Technology, Poland

**Jerzy Grzymala-Busse,** University of Kansas, USA

**Jakub Gutenbaum,** Systems Research Institute of the Polish Academy of Sciences, Poland

**Zdzisław Hippe,** University of Information Technology and Management in Rzeszow, Poland

**Jerzy W. Jaromczyk,** University of Kentucky, USA

**Piotr Jedrzejowicz,** Gdynia Maritime University, Poland

**Yaochu Jin,** Honda Research Center, Germany

**Janusz Kacprzyk,** Systems Research Institute of the Polish Academy of Sciences, Poland

**Etienne E. Kerre,** Ghent University, Belgium

**Mieczysław A. Kłopotek,** Institute of Computer Science, PAS, Poland

**Waldemar Koczkodaj,** Laurentian University, Canada

**Mieczyslaw Kokar,** Northeastern University, USA

**Jozef Korbicz,** University of Zielona Gora, Poland

**Witold Kosinski,** Polish Japanese Institute of Information Technology, Poland

**Adam Krzyzak,** Concordia University, Canada

**Casimir Kulikowski,** Rutgers University. USA

**Juliusz Kulikowski,** Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences, Poland

**Jacek Mandziuk,** Warsaw University of Technology, Poland

**Victor W. Marek,** University of Kentucky, USA

**Zbigniew Michalewicz,** University of Adelaide, Australia

**Zbigniew Nahorski,** Systems Research Institute of the Polish Academy of Sciences, Poland

**Daniel Neagu,** University of Bradford, UK

**Hung Son Nguyen,** The University of Warsaw, Poland

**Sankar Pal,** Indian Statistical Institute, India

**Witold Pedrycz,** University of Alberta, Canada

**James F. Peters,** University of Manitoba, Canada

**Adam Przepiorkowski,** Institute of Computer Science, Polish Academy of Sciences, Poland

**Zbigniew Ras,** University of North Carolina, USA

**Danuta Rutkowska,** Czestochowa University of Technology, Poland

**Leszek Rutkowski,** Czestochowa University of Technology, Poland

**Khalid Saeed,** Bialystok Technical University, Poland

**Abdel-Badeh M Salem,** Ain Shams University, Egypt

**Franciszek Seredynski,** Institute of Computer Science of the Polish Academy of Sciences, Poland

**Andrzej Skowron,** The University of Warsaw, Poland

**Roman Slowinski,** Poznan University of Technology, Poland

**Zbigniew Suraj,** Rzeszów University, Poland

**Jerzy Swiatek,** Wroclaw University of Technology, Poland

**Andrzej Szalas,** The University of Warsaw, Poland

**Piotr Szczepaniak,** Technical University of Lodz, Poland

**Stan Szpakowicz,** University of Ottava, Canada

**Ryszard Tadeusiewicz,** AGH University of Science and Technology, Poland

**Alicja Wakulicz-Deja,** University of Silesia, Katowice, Poland

**Guoyin Wang,** Nankai University, China

**Anita Wasilewska,** Stony Brook University, Ny, Usa

**Slawomir T. Wierzchon,** Institute of Computer Science, PAS, Poland

**Bogdan M. Wilamowski,** Auburn University, USA

**Ning Zhong,** Maebashi Institute of Technology, Japan

**Wojciech Ziarko,** University of Regina, Canada

**Jacek M. Zurada,** University of Louisville, USA

### Organizing Committee

**Halina Kwasnicka, Urszula Markowska-Kaczmar,** Institute of Applied Informatics, Wroclaw University of Technology, Wroclaw, Poland

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 65 – 71

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Applying Differential Evolution to the Reporting Cells Problem

Sónia Almeida-Luz
Polytechnic Institute of Leiria,
School of Technology and Management
Leiria, Portugal
sluz@estg.ipleiria.pt

Miguel A.Vega-Rodríguez,
Juan A. Gómez-Pulido,
Juan M. Sánchez-Pérez
University of Extremadura, Dept. Technologies
of Computers and Communications
Escuela Politécnica, Campus Universitario
s/n 10071 Cáceres, Spain
{mavega, jangomez,sanperez}@unex.es

*Abstract*—**The Location Management problem corresponds to the management of the network configuration with the objective of minimizing the costs involved. It can be defined using several different schemes that principally consider the location update and the paging costs. The Location Area and Reporting Cells are two common strategies used to solve the location management problem. In this paper we present a new approach that uses the Differential Evolution algorithm applied to the reporting cells planning problem, with the objective of minimizing the involved location management costs. With this work we want to define the best values to the differential evolution parameters and respective scheme, using 12 distinct test networks, as well as compare our results with the ones obtained by other authors. The final results obtained with this approach are very encouraging.**

## I. Introduction

THE use of mobile networks is growing every day and being applied to the most of newly and renovated applications for data transfer, voice and fax services among many others mobile services. Because of this, communication networks [1] must support a big number of users and their respective applications maintaining a good response without loose quality and availability. With the goal that mobile networks keep this quality it is necessary to consider the mobility management when making design of the network infrastructure.

Mobility management is a very important point because it includes the process of hand off management that enables the mobile network to locate roaming mobile terminals, and also the process of location management that enables the mobile network to find the current location of a mobile terminal in order to make or receive calls. We are mainly concerned with location management because it involves the user movements and tracing, with the objective of minimizing the involved costs.

The location management is partitioned in two main operations: location update that corresponds to the notification of current location, performed by mobile terminals when they change their location in the mobile network, and location paging (inquiry) that represents the operation of determining the location of the mobile user terminal, performed by the network when it tries to direct an incoming call to the user.

There exist several strategies of location management, which are divided in two main groups: static and dynamic schemes [2]. The static schemes consider the same behavior of the network for all users, while the dynamic schemes consider different network topologies for different users based on the individual user's call and mobility patterns. A survey of different dynamic techniques based on users' behavior such as timer-based, distance-based, movement-based (among others) may be seen in [2]. As static techniques, the most common ones are always-update, never-update, and location area schemes [2], [3], among others. The reporting cells (RC) represents another static location management strategy that also is very used.

In this paper we present a new approach to solve the reporting cells problem and minimize the involved costs, using the Differential Evolution (DE) based algorithm. Section II presents an overview of the reporting cells planning and the respective involved costs. In section III, the DE based algorithm is described including its parameters and respective possible schemes. In section IV are explained the details of algorithm implementation and network preparation. Section V includes the experimental results and presents the results produced by the four distinct experiments. In section VI analysis and comparisons with other authors' results and other artificial life techniques are shown. Finally section VII includes conclusions and future work.

## II. Reporting Cells Planning Problem

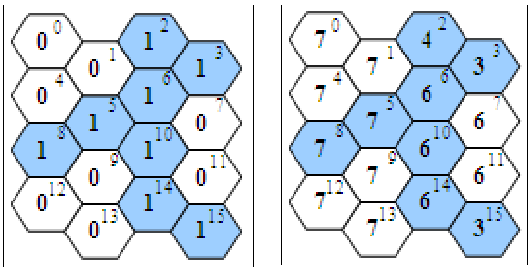
In this section we will explain the reporting cells scheme and how it is applied in the calculus of the location management cost.

### A. Reporting Cells Scheme

The reporting cells planning scheme was proposed by Bar and Kessler in [4] with the objective of minimizing the cost of tracking mobile users.

This strategy is characterized by defining a subset of cells as reporting cells and the others as non-reporting cells (nRC), as it is possible to see in Fig. 1a (RC represented with value 1 and in blue color and nRC represented with value 0 and in white color). The mobility terminals only perform a

978-83-60810-14-9/08/$25.00 © 2008 IEEE



a) b)

Fig 1. a) Reporting Cells planning b) Vicinity values

new location update when they change their location and move to one reporting cell. If an incoming call must to be routed to the mobile user, the search can be restricted to his last reporting cell known and their respective neighbors non-reporting cells.

It is necessary to calculate for each cell the vicinity factor, which represents the maximum number of cells that the user must page when an incoming call occurs.

The vicinity value of a reporting cell corresponds to the number of non-reporting cells that are reachable from this reporting cell, without crossing other reporting cells, and adding the reporting cell itself. For example, considering the calculus of vicinity factor for the cell number 5 (RC) in Fig. 1a, we must count the number of neighbors that are nRCs (cells 0, 1, 4, 9, 12 and 13) and also include the RC itself, which makes a total of 7 neighbors. This total number of neighbors will correspond to the vicinity factor of this RC.

If we are calculating the vicinity value of a non-reporting cell it is necessary to consider the maximum vicinity value among the reporting cells from where this one can be reached. This means that if a non-reporting cell belongs to the neighborhood of more than one reporting cell, the calculus has to be done for all the reporting cells and then, the maximum number is set as the vicinity factor of the respective non-reporting cell. If we consider the cell number 9 (nRC), in Fig. 1a, we can observe that it belongs to the neighborhood of at least two RCs (more precisely four cells: number 5, 8, 10 and 14). Because of that, the calculus of vicinity factor must be done for all those RCs and after this, the maximum number will be considered as the vicinity factor for this nRC. The vicinity factor for cells 5, 8, 10 and 14 is respectively 7, 7, 6 and 6, so the maximum value that represents the vicinity factor of cell number 9 is 7.

Considering the reporting cells planning of the Fig. 1a and calculating all the vicinity factors, the result will be the one presented on Fig. 1b.

### *B. Location Management Cost*

The location management (LM) cost is principally divided in two fundamental operations of location update and location paging. The location update (LU) cost corresponds to the cost involved with the location updates performed by mobile terminals in the network, when they change their location and must register the new one. The location paging (P) cost is caused by the network when it tries to locate a user's mobile terminal, during the location inquiry, and normally the number of paging transactions is directly related to the number of incoming calls. LM involves several other parameters and components that are considered to be equal for all strategies and does not make influence when comparing the results obtained by different strategies. Because of that these cost are not considered for the total cost.

From earlier studies and experiments [3], [5] we have seen that the generic formula to obtain the LM cost is:

$$Cost=\beta * N_{LU}+N_{P} \qquad (1)$$

The total cost of location updates is given by $N_{LU}$, the total cost of paging operations corresponds to $N_P$ and $\beta$ is a ratio constant used in a location update relatively to a paging transaction in the network. To each location update is imputed a much higher cost than to each paging operation, because of the complex process that must be executed for each location update performed, and also because most of the time a mobile user moves without making any call [6]. Due to all of that, the cost of a location update is normally considered to be 10 times greater than the cost of paging, that is, $\beta$=10 [5].

In the reporting cells scheme the location updates only are performed when a mobile user enters in a reporting cell and the vicinity factor of each cell must be considered. Because of that the generic formula given by (1) must be readjusted and it is formulated as [7]:

$$Cost=\beta * \sum_{i\in s} N_{LU}(i)+\sum_{i=0}^{N} N_{P}(i) * V(i) \qquad (2)$$

Here we can see that $N_{LU}(i)$ corresponds to the number of location updates associated to the reporting cell *i*, *S* indicates the subset of cells defined as reporting cells, $N_P(i)$ is the number of incoming calls attributed for cell *i*, *N* is the total number of cells that compound the mobile network configuration and *V(i)* is the vicinity factor attributed for cell *i*.

We will use this formula with the objective of minimize the LM cost, using the reporting cells strategy.

## III. Differential Evolution

The Differential Evolution (DE) is a population-based algorithm, created by Ken Price and Rainer Storn [8], whose main objective is functions optimization. This algorithm is one strategy based on evolutionary algorithms that has some specific characteristics. It has a key strategy to generate new individuals by calculating vector differences between other randomly-selected individuals of the population.

### *A. DE Parameters*

DE algorithm uses four important parameters: population size *NI*, mutation *F*, crossover *Cr* and selection operators as well as different schemes presented in the following sub-section. For further information about the four parameters, refer to [8].

### *B. DE Schemes*

DE can be implemented using 10 different schemes suggested by Price and Storn [8]. These schemes, exposed in Table I, are classified based on notation *DE/x/y/z*, where *x* specifies the vector to be mutated, *y* correspond to the number of difference vectors used in mutation of *x* (normally 1 or 2) and *z* represents the crossover scheme. The vector *x* may be chosen randomly ('rand') or as the best of current popula-

TABLE I. DE SCHEMES

| Scheme | Mutant vector generation |
|---|---|
| DE/best/1/exp | $x_i = x_{best} + F\,(x_{r1} - x_{r2})$ |
| DE/rand/1/exp | $x_i = x_{r3} + F(x_{r1} - x_{r2})$ |
| DE/randtobest/1/exp | $x_i = x_{r3} + F(x_{best} - x_{r3}) + F(x_{r1} - x_{r2})$ |
| DE/best/2/exp | $x_i = x_{best} + F\,(x_{r1} + x_{r2} - x_{r3} - x_{r4})$ |
| DE/rand/2/exp | $x_i = x_{r5} + F\,(x_{r1} + x_{r2} - x_{r3} - x_{r4})$ |
| DE/best/1/bin | $x_i = x_{best} + F\,(x_{r1} - x_{r2})$ |
| DE/rand/1/bin | $x_i = x_{r3} + F(x_{r1} - x_{r2})$ |
| DE/randtobest/1/bin | $x_i = x_{r3} + F(x_{best} - x_{r3}) + F(x_{r1} - x_{r2})$ |
| DE/best/2/bin | $x_i = x_{best} + F\,(x_{r1} + x_{r2} - x_{r3} - x_{r4})$ |
| DE/rand/2/bin | $x_i = x_{r5} + F\,(x_{r1} + x_{r2} - x_{r3} - x_{r4})$ |

tion ('best') and *z* may be binomial ('bin') or exponential ('exp') depending on the type of crossover used.

### C. DE Algorithm

The pseudo-code of the DE algorithm, using the *DE/best/1/exp* is presented in Fig. 2. It starts by defining and evaluating the initial population through calculating the fitness value for each individual. After that, until the termination condition is not reached, the necessary individuals are picked and a new one is produced according to the selected DE scheme and respective rules. This new individual is evaluated and compared with the old one. Just the one with the best fitness value will be chosen and pass for population of the next generation.

## IV. IMPLEMENTATION DETAILS

In this section we explain the fitness function implemented to evaluate the solutions obtained, present the test networks used and expose the original definition of parameters.

```
Initialize the population
Evaluate the initial population
While (termination condition not satisfied) {
        Randomly select ind. $xr1 \neq xbest$
        Randomly select ind. $xr2 \neq xr1$ and $\neq xbest$
        Generate trial ind.: $xtrial = xbest + F(xr1 - xr2)$
        Use Cr to define the amount of genes changed
            in trial individual
        Evaluate the trial individual
        Deterministic selection
}
```

Fig 2. DE Algorithm Pseudo Code with Scheme *DE/best/1/exp*

### A. Fitness Function

In this study the fitness function is used for measuring the total location management cost of each potential solution, which is defined according to the equation (2). This means that for each potential solution generated, it is calculated the fitness value, which corresponds to the network configuration by means of reporting cells and non-reporting cells.

### B. Test Networks

Other authors have studied the reporting cells strategy, but most of them do not present the test networks used, so it is not possible to compare our approach with them. However, in [7] it is presented a set of twelve test networks, representing four groups defined by size, that have been generated and are available in [9] as benchmark. In this work we used these twelve test networks with the objective of compare final results. In Table II it is shown, as an example, the test network 1 that represents a 4x4 cells configuration. The first column indicates the cell identification, the second column corresponds to the number of location updates *NLU* and the third represents the number of incoming calls *NP*.

### C. Parameters Definition

The initial definition of parameters is an important step because it represents the basis for the algorithm evolution. First it is defined the initial population of candidate solutions that corresponds to the individuals.

Each individual is compound by *N* genes, where the *N* value is the number of cells in the network and each gene represents the information about the cell type, which can be a reporting cell or a non-reporting cell.

To define the initial population we have set, with a probability of fifty percent, the type of each cell as RC or nRC.

Initially it is also necessary to set the DE algorithm parameters and that has been done with a number of individuals *NI* equal to 10, the mutation factor *F* set to 0.5 and the crossover value *Cr* defined as 0.1. For the DE scheme it has been selected the *DE/rand/1/bin*. The number of generations represents the terminal condition when the algorithm is executed and it is set to 1000.

Throughout the different experiments, the parameters values have been adjusted with the specific objective of obtaining the best results for each test network.

TABLE II. TEST NETWORK 1

| Cell | NLU | NP | Cell | NLU | NP |
|---|---|---|---|---|---|
| **0** | 452 | 484 | **8** | 647 | 366 |
| **1** | 767 | 377 | **9** | 989 | 435 |
| **2** | 360 | 284 | **10** | 1105 | 510 |
| **3** | 548 | 518 | **11** | 736 | 501 |
| **4** | 591 | 365 | **12** | 529 | 470 |
| **5** | 1451 | 1355 | **13** | 423 | 376 |
| **6** | 816 | 438 | **14** | 1058 | 569 |
| **7** | 574 | 415 | **15** | 434 | 361 |

TABLE III. EXPERIMENT 1: DETERMINING THE BEST NI

| Test Network | NI – Fitness Evaluation | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| N. (Dim) | 10 | 25 | 50 | 75 | 100 | 125 | 150 | 175 | 200 | 225 |
| 1 (4x4) | 99,137 | 100,881 | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 |
| 2 (4x4) | 101,250 | 98,879 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 |
| 3 (4x4) | 98,106 | 101,403 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 |
| 4 (6x6) | 195,283 | 185,092 | 176,800 | 173,701 | 173,701 | 173,701 | 173,701 | 173,701 | 173,701 | 173,701 |
| 5 (6x6) | 205,859 | 192,426 | 185,937 | 182,331 | 182,331 | 182,331 | 182,331 | 182,331 | 185,059 | 182,331 |
| 6 (6x6) | 193,540 | 186,423 | 175,321 | 174,519 | 174,519 | 174,519 | 174,519 | 174,519 | 174,519 | 174,519 |
| 7 (8x8) | 338,196 | 321,575 | 315,097 | 310,888 | 308,853 | 309,342 | 308,401 | 308,401 | 308,991 | 308,401 |
| 8 (8x8) | 319,912 | 304,750 | 294,548 | 287,149 | 289,051 | 287,149 | 287,149 | 287,149 | 289,935 | 287,149 |
| 9 (8x8) | 292,467 | 276,299 | 270,171 | 265,272 | 264,204 | 264,204 | 264,204 | 264,316 | 264,204 | 264,204 |
| 10 (10x10) | 425,866 | 405,127 | 394,168 | 388,206 | 386,775 | 387,551 | 386,695 | 386,474 | 387,543 | 386,893 |
| 11 (10x10) | 390,793 | 377,363 | 361,299 | 360,210 | 361,581 | 359,224 | 358,778 | 359,224 | 359,697 | 358,944 |
| 12 (10x10) | 401,704 | 385,022 | 380,846 | 375,233 | 376,631 | 374,711 | 375,001 | 375,722 | 373,733 | 374,220 |

## V. EXPERIMENTAL RESULTS

In this section we explain the four distinct experiments applied to each test network with the objective of study in more detail the best configuration of DE. For each experiment, and for all combination of parameters, 30 independent runs have been performed in order to assure its statistical relevance. In each experiment the final results, of the best fitness values obtained (lower cost value), are presented and explained the decisions taken.

### *A. Experiment 1 – Determining NI*

The number of individuals that will compound the initial population must be the first experiment because it is the basis of the algorithm implementation. In order to accomplish that, we have fixed, as referred in *IV.C. Parameters Definition,* the values of crossover *Cr*=0.1, the mutation *F*=0.5, DE scheme as *DE/rand/1/bin* and the stop criterion as 1000 generations, considering our experience from earlier experiments that we have performed [3].

With this experiment we have concluded that, increasing of *NI* value, it is possible to observe a positive evolution of the results (see fitness results in Table III ), but just until the value of *NI* =175, because after that we start observing worse results and stop increase in *NI* =225. Considering this and the average evolution we concluded that *NI* =175 would be the elected value for the second experiment.

### *B. Experiment 2 – Determining Cr*

The second experiment has the objective of selecting the *Cr* value that obtains the best results. To proceed with this experiment we fixed the value of *NI* to 175 (from experiment 1), and maintained the other parameters as defined in the beginning of experiment 1.

This experiment has been executed using different values for *Cr*: 0.1, 0.25, 0.50, 0.75 and 0.90. Analyzing the results obtained, we could conclude that best values were obtained with *Cr*=0.1 and *Cr*=0.25. Because of that, with the objective of taking more complete conclusions, we decided to execute the algorithm with values 0.15 and 0.20. Finally, as it is possible to see in , we could conclude that *Cr*=0.15 is the one that performs better.

TABLE IV. EXPERIMENT 2: DETERMINING THE BEST CR

| Test Network | Cr – Fitness Evaluation | | | | | | |
|---|---|---|---|---|---|---|---|
| N. (Dim) | 0.1 | 0.15 | 0.20 | 0,25 | 0.50 | 0.75 | 0.90 |
| 1 (4x4) | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 | 98,535 |
| 2 (4x4) | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,258 |
| 3 (4x4) | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 98,216 |
| 4 (6x6) | 173,701 | 173,701 | 173,701 | 173,701 | 173,701 | 177,647 | 177,889 |
| 5 (6x6) | 182,331 | 182,331 | 187,990 | 183,264 | 184,679 | 183,991 | 185,966 |
| 6 (6x6) | 174,519 | 174,519 | 174,519 | 175,321 | 175,182 | 175,321 | 178,255 |
| 7 (8x8) | 308,401 | 308,401 | 308,401 | 311,646 | 313,378 | 313,607 | 319,069 |
| 8 (8x8) | 287,149 | 287,149 | 289,573 | 289,051 | 293,248 | 302,812 | 309,609 |
| 9 (8x8) | 264,204 | 264,204 | 265,452 | 264,786 | 272,249 | 266,876 | 275,489 |
| 10 (10x10) | 387,318 | 386,681 | 388,357 | 386,959 | 393,510 | 393,492 | 420,650 |
| 11 (10x10) | 360,262 | 358,669 | 360,072 | 360,128 | 360,596 | 367,508 | 374,405 |
| 12 (10x10) | 373,695 | 374,966 | 374,554 | 374,921 | 377,190 | 383,782 | 391,001 |

TABLE V. EXPERIMENT 3 - DETERMINING THE BEST F

| Test Network | F – Fitness Evaluation | | | | |
|---|---|---|---|---|---|
| N. (Dim) | 0.1 | 0,25 | 0.50 | 0.75 | 0.90 |
| 1 (4x4) | 98,535 | 98,535 | 98,535 | 98,535 | 98,727 |
| 2 (4x4) | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 |
| 3 (4x4) | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 |
| 4 (6x6) | 174,112 | 173,701 | 173,701 | 173,701 | 176,530 |
| 5 (6x6) | 182,331 | 182,331 | 182,331 | 182,331 | 182,331 |
| 6 (6x6) | 174,519 | 175,321 | 174,519 | 174,519 | 174,519 |
| 7 (8x8) | 310,162 | 310,426 | 308,401 | 311,492 | 308,401 |
| 8 (8x8) | 293,093 | 304,911 | 287,149 | 292,913 | 295,557 |
| 9 (8x8) | 265,494 | 264,643 | 264,204 | 268,312 | 265,750 |
| 10 (10x10) | 388,849 | 389,438 | 386,681 | 389,125 | 387,533 |
| 11 (10x10) | 359,221 | 360,072 | 358,669 | 358,167 | 361,441 |
| 12 (10x10) | 373,298 | 375,087 | 374,966 | 371,829 | 375,232 |

### C. *Experiment 3 – Determining F*

The determination of the best value for mutation, *F*, is the purpose of the third experiment. So, in order to perform this experiment it was fixed the value of *NI* to 175 (from experiment 1), the value of *Cr* to 0.15 (from experiment 2) and the others maintained as in the two earlier experiments.

After finishing these executions and examining the results (see Table V) we conclude that *F*=0.5 is the one that permits to obtain the best results.

### D. *Experiment 4 – Determining DE Scheme*

Finally, with this fourth experiment we pretend to select the most adequate DE scheme, the one that permits to obtain the best results (lower fitness value). For that, we fixed the best values for each parameter (defined in the three earlier experiments) as: *NI*=175, *Cr*=0.15 and *F*=0.5, and executed the algorithm applying the ten DE schemes presented in section *III.B*.

Once finished all the executions, and observing the respective results shown in Table VI, it was possible to conclude that the scheme *DE/rand/1/bin* is the one with a better performance, because it is the one that obtains better fitness values for all the test networks. With these results we may say that the binomial schemes perform better than the exponential ones and that it is also better to choose randomly the individuals used to create the trial individual.

Finishing these four experiments we had determined the best DE configuration, applied to the reporting cells plan ning problem, setting the parameters as *NI* =175, *Cr* =0.15, *F* =0.5 and *DE/rand/1/bin* as the most adequate DE scheme.

## VI. ANALYSIS AND COMPARISONS

In this section we pretend to analyze the results obtained, compare them with those shown in [7] and present the configuration for best solutions.

Finally we apply the best DE configuration to the test networks presented in [10] and compare results produced with this DE configuration, with the ones of other artificial life techniques.

### A. *Analysis and Comparison of Results*

Analyzing the experimental results we could conclude that with this approach it is possible to obtain the same minimum fitness values (considered optimal in [7]), as the ones obtained in [7] with a Hopfield Neural Network with Ball Dropping (HNN+BD) and a Geometric Particle Swarm Optimization (GPSO) for ten of the twelve test networks used.

For the other two test networks the results are very similar because: for the test-network-10 our fitness value is 386,951 and in [7] the one obtained by the HNN+BD is 386,351; and for the test-network-12 our fitness value is 371,829 and in [7] the value obtained by HNN+BD and GPSO is 370,868. Relatively to the average values it is possible to say that they are very similar.

In fig. 3 the configuration for each test-network solution is shown and it is possible to observe that most of them split each one in subnetworks.

### B. *Comparison with other Artificial Life Techniques*

After obtaining the best DE configuration applied to the reporting cells planning problem we decide to apply it in

TABLE VI. EXPERIMENT 4: DETERMINING THE DE SCHEME

| Test Network | DE Scheme – Fitness Evaluation | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Exponential Crossover | | | | | Binomial Crossover | | | | |
| N. (Dim) | Best1 | Rand1 | RandTBest1 | Best2 | Rand2 | Best1 | Rand1 | RandTBest1 | Best2 | Rand2 |
| 1 (4x4) | 98,535 | 98,535 | 98,535 | 98,535 | 99,008 | 98,535 | 98,535 | 98,535 | 98,727 | 98,535 |
| 2 (4x4) | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 | 97,156 |
| 3 (4x4) | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 | 95,038 |
| 4 (6x6) | 173,701 | 173,701 | 173,701 | 176,530 | 178,038 | 176,041 | 173,701 | 173,701 | 173,701 | 173,701 |
| 5 (6x6) | 182,331 | 182,331 | 187,801 | 190,779 | 191,279 | 182,331 | 182,331 | 182,331 | 182,331 | 182,331 |
| 6 (6x6) | 174,519 | 175,182 | 183,992 | 177,276 | 177,892 | 181,850 | 174,519 | 174,519 | 174,519 | 174,519 |
| 7 (8x8) | 322,973 | 319,772 | 328,327 | 323,391 | 332,472 | 320,236 | 308,401 | 308,401 | 309,855 | 308,730 |
| 8 (8x8) | 304,214 | 307,139 | 313,010 | 310,708 | 316,849 | 305,236 | 287,149 | 287,149 | 287,149 | 287,149 |
| 9 (8x8) | 277,408 | 279,177 | 290,646 | 291,684 | 289,936 | 269,984 | 264,204 | 264,316 | 265,164 | 264,353 |
| 10 (10x10) | 420,701 | 423,017 | 420,452 | 421,353 | 425,228 | 394,176 | 386,681 | 386,951 | 393,471 | 386,695 |
| 11 (10x10) | 385,950 | 380,824 | 384,661 | 387,273 | 380,241 | 366,156 | 358,167 | 359,486 | 367,202 | 359,517 |
| 12 (10x10) | 394,636 | 388,468 | 395,290 | 395,404 | 394,767 | 379,227 | 371,829 | 376,015 | 379,544 | 376,165 |

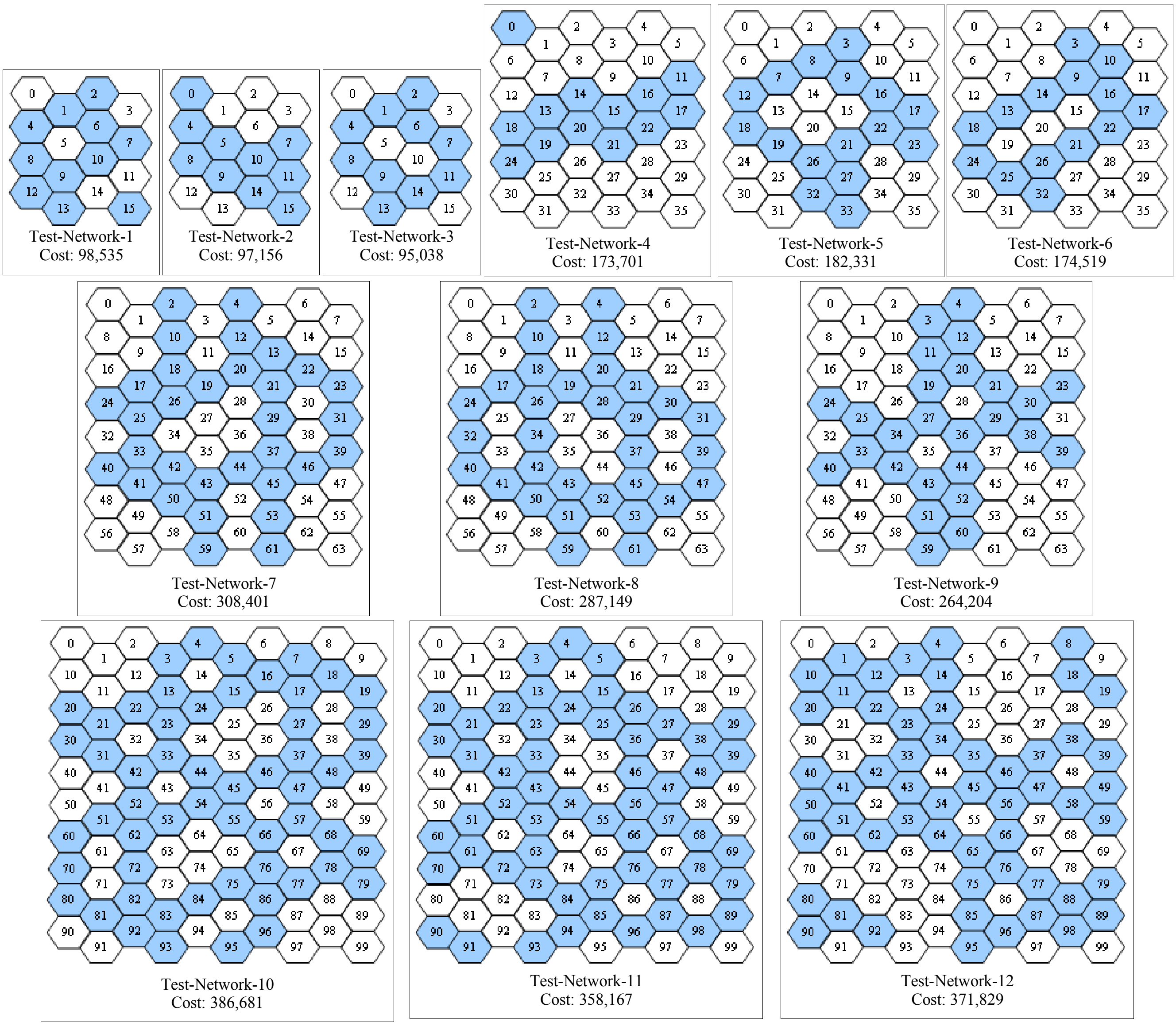


*Fig 3. Test-Network Solution with Reporting Cells Configuration*

other test networks, and analyze the performance with the objective of comparing with other artificial life techniques. We decided to use 3 test networks presented in [10] (also referred in [7]) in order to compare our results with those produced by the application of Genetic Algorithms (GA), Tabu Search (TS) and Ant Colony algorithm (AC).

Making the additional experiments we conclude that our approach performs well. Using the Test-Network-1 (4×4 instance provided in [10]) we obtain the same fitness value as the GA, TS and AC, that is, 92,883 with a total of 10 reporting cells.

Our approach generates a better solution when solving the Test-Network-2 (6×6 instance provided in [10]), comparing with GA and AC. The fitness value obtained by the GA is 229,556 with a total of 26 reporting cells in the network [7], [10], while, the cost obtained by DE in this work is 211,278 with 24 reporting cells. TS presents the same cost 211,278 and the fitness value obtained by AC is 211,291.

Finally, for the Test-Network-3 (8×8 instance in [10]) DE again surpasses the results obtained in [10] by the GA, TS and AC. Concretely, the fitness value obtained by the GA and TS is 436,283, the one obtained by AC is 436,886 while, the cost obtained by DE in this work is 436,269 with a total of 39 reporting cells.

## VII. Conclusion

In this paper we have discussed the use of differential evolution algorithm (DE) applied to the reporting cells planning problem. To the best of our knowledge, this is the first time that DE is employed for this task (this is another contribution of this paper).

We have studied in detail the best configuration of DE including parameters and scheme. After more than 10,000 runs, they are *NI*=175, *Cr*=0.15, *F*=0.5 and *DE/ran/1/bin* as the best DE scheme.

We have shown that our approach produces interesting results because when compared with ones of other authors, that use HNN+BD and GSPO, they are equal or very similar.

Comparing the performance of DE algorithm with other artificial life techniques as genetic algorithm (GA), tabu search (TS), and ant colony algorithm (AC), we may say that it performs well because improves the results obtained by those ones.

As future work we pretend applying other evolutionary algorithms to the RC problem comparing their results with the ones accomplished by the DE algorithm. We also have the intention of testing our approach with test networks generated by using real data.

## Acknowledgment

This work was developed in part thanks to the project OPLINK (TIN2005-08818-C04-03). Thanks also to the Polytechnic Institute of Leiria, for the economic support offered to Sónia Almeida-Luz to make this research.

## References

[1] K. Pahlavan, A. H. Levesque, *Wireless Information Networks* , John Wiley & Sons, Inc, 1995.

[2] V. W. S. Wong, V. C. M. Leung, “Location Management for Next-Generation Personal Communications Networks”, IEEE Network, Oct. 2000, vol. 14, no. 5, pp. 18-24.

[3] Almeida-Luz, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, J. M. Sánchez-Pérez, “Defining the Best Parameters in a Differential Evolution Algorithm for Location Area Problem in Mobile Networks”, In EPIA 2007 – 13th Portuguese Conference on Artificial Intelligence; (Eds). APPIA, Dec. 2007, ISBN: 978-98-995-6180-9; pp. 219-230.

[4] A. Bar-Noy, I. K. “Tracking mobile users in wireless communications networks”, *IEEE Transactions on Information Theory* , 1993, vol. 39, pp. 1877 – 1886.

[5] J. Taheri, A. Y. Zomaya, “A Genetic Algorithm for Finding Optimal Location Area Configurations for Mobility Management”, *30th Anniversary of the IEEE Conference on Local Computer Networks (LCN)*, Nov. 2005, pp. 568-577.

[6] P. R. L. Gondim, “Genetic Algorithms and the Location Area Partitioning Problem in Cellular Networks”, *IEEE 46th Vehicular Technology Conf. Mobile Technology for the Human Race* , May 1996, vol. 3, pp. 1835-1838.

[7] E. Alba, J. García-Nieto, J. Taheri, A. Zomaya, “New Research in Nature Inspired Algorithms for Mobility Management in GSM Networks”, EvoWorkshops 2008, LNCS 4974, pp. 1-10, Springer-Verlag Berlin Heidelberg, 2008.

[8] Price, R. Storn: Web Site of Differential Evolution: http://www.icsi.berkeley.edu/~storn/code.html (on June 2008)

[9] Test networks benchmark: http://oplink.lcc.uma.es/problems/mmp.html (on June 2008)

[10] R. Subrata, A. Y. Zomaya, “A comparison of three artificial life techniques for reporting cell planning in mobile computing”, IEEE Trans. Parallel Distrib, Syst. 14(2), 142–153, 2003.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 73–78

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Towards Word Sense Disambiguation of Polish

Dominik Baś, Bartosz Broda, Maciej Piasecki
Institute of Applied Informatics, Wrocław University of Technology, Poland
Email: {bartosz.broda, maciej.piasecki}@pwr.wroc.pl

***Abstract*—We compare three different methods of Word Sense Disambiguation applied to the disambiguation of a selected set of 13 Polish words. The selected words express different problems for sense disambiguation. As it is hard to find works for Polish in this area, our goal was to analyse applicability and limitations of known methods in relation to Polish and Polish language resources and tools. The obtained results are very positive, as using limited resources, we achieved the accuracy of sense disambiguation greatly exceeding the baseline of the most frequent sense. For the needs of experiments a small corpus of representative examples was manually collected and annotated with senses drawn from plWordNet. Different representations of context of word occurrences were also experimentally tested. Examples of limitations and advantages of the applied methods are discussed.**

## I. Introduction

POLYSEMY of word forms seems to be an intrinsic feature of the natural language and can be observed in any natural language including Polish. It is a stumbling block for semantic text processing and complicates access to meanings in a semantic lexicon. One needs an algorithm choosing the most appropriate meaning of the given word form in relation to the given context. This need was noticed as early as in 50s, and continuous works on *Word Sense Disambiguation* (henceforth abbreviated to WSD) have been performed since 70s, e.g. the *Word Expert* system of Wilks [1].

Research on WSD has been conducted for English but also for many other languages. Despite its importance, it is very hard to find published works, working systems or practical results for Polish. Development of the corpus annotated by word senses and next development of a WSD algorithm is planned as a part of the project on the construction of the National Corpus of Polish [2].

Our main objective is to developed a robust WSD method for Polish which will be based on the Polish wordnet called *plWordNet* [3], as the source of lexical meanings. However, as such a far going enterprise requires a lot of time and workload, we decided to start with a much more limited experiment. The goal of the work presented here was to develop a WSD algorithm for selected Polish words going in the line of *lexical sample* task of Senseval Evaluation Exercises [4]. Also, lexical sample task is one approach to minimise the utilised resources, especially manual work.

As we wanted to achieve a relatively high accuracy, from the very beginning we assumed a supervised learning model and a construction of a small training corpus, in which selected words were manually annotated with plWordNet synsets. In the case of the supervised learning algorithms one can control which senses are learned and identified in text. Nevertheless, research on unsupervised sense discovery have been also performed in parallel, e.g. [5], [6].

## II. Construction of the Training Corpus

As none of the existing Polish corpora has been semantically annotated, we decided to select subcorpus of the IPI PAN Corpus (IPIC) [7] and extend it with annotation by word senses for occurrences of chosen words. We selected 13 different base word forms corresponding to several polysemous lexemes and homonyms. Choosing the subset, we tried to represent the variety of different problems for WSD. We selected polysemous lexemes possessing from 2 to 7 senses, where some of the senses have homonymous character, i.e. those senses represent separate homonyms of the same morphological base form. We included homonyms, as they express very different meanings and it should be easy to differentiate between their senses. The important factor in selection was also the coverage of plWordNet, which still does not describe some rare senses. Finally, we also tried to compose the set of words as consisting of words that are intuitively less or more difficult for possible WSD algorithms. The selected set includes:

1) ***agent*** (4 senses, English translation: *agent*),
2) ***automat*** (3: *automaton*, *automatic machine* and *machine-gun*),
3) **dziób** (4: *beak*, *bow*, *mouth* (semantically marked) and *face* (semantically marked)),
4) ***język*** (2: *language* or *tongue*),
5) ***klasa*** (6: *class*, *classroom*, *form*, *grade*),
6) ***linia*** (6: *line*, *route*, *figure*, *contour*),
7) ***pole*** (3: *field*),
8) ***policja*** (2: *police*, *police station*),
9) ***powód*** (2: *reason* and *protection* (person)),
10) ***sztuka*** (6: *art*, *play*, *craft*, *item*),
11) ***zamek*** (4: *castle*, *door lock*, *zipper* and *breechblock*),
12) ***zbiór*** (7: *set*, *collection*, *harvest* and *file*),
13) ***zespół*** (5: *team*, *gruop*, *band*, *company*, *complex* and *unit*).

Henceforth the selected base word forms will be called *training words*.

For sense inventory we used plWordNet - a resource under development, but it has already reached the size of 14 677 lexical units and is publicly available for research [3]. plWordNet follows the general scheme of the Princeton WordNet but it was constructed from scratch in bottom-up way. The starting point was the list of about 10 000 most frequent word base forms from IPIC. As a result plWordNet includes the

978-83-60810-14-9/08/$25.00 © 2008 IEEE

most frequent and most general Polish words and multiword lexical units (some of them were added by lexicographers during construction). In a way typical for wordnets, plWordNet introduces a fine grained distinction of senses.

IPIC is the only available large, morphosyntactically annotated corpus of Polish and consists of about 254 millions of tokens. We started construction of the semantically annotated subcorpus with finding all IPIC documents including at least one of the training words. Next, the documents had to be manually inspected in order to balance the number of examples for different training words, and their occurrences representing different senses. As the frequencies of senses vary in large extent, it was necessary to analyse hundreds of training word occurrences in order to find an example of some rare sense. For some sense it was difficult to find enough examples, e.g. the training word *dziób* occurs 536 times in IPIC, but after the manual inspection of all cases we found that the emotionally marked sense of *dziób* meaning *mouth* can be spotted only 9 times. Our experiments confirm phenomenon observed for English [8]—even for general and frequent words some senses are underrepresented even in a large corpora.

While selecting documents for the training subcorpus we took into account their genres and origins. The subcorpus consists of literature works, press articles and news, scientific works and legal texts. We paid special attention to avoiding situations in which all examples for some sense would be taken from the same source text. It could negatively bias the disambiguation process, as some characteristic or even idiosyncratic properties of the given text (e.g. originating from the style of the given author) could be learned by the disambiguator.

Annotation of the subcorpus was done with the help of the modified version of the annotation editor called *Manufakturzysta* [9] constructed especially for the IPIC XML format [7] based on the XCES general format. Occurrences of training words were annotated with synset identifiers from plWordNet. Annotation was performed mainly by one of the co-authors. However, in many difficult cases in which the appropriate assignment of a sense to word occurrence was unclear, the other two co-authors were consulted. The final decision often was difficult as plWordNet is still under construction and there was always possibility that some sense of the given training word is not described in plWordNet yet.

Collocations, like *pole chwały* (*field of glory*) or *ugryźć się w język* (*to bite one's tongue*), appeared to be a problem. They should be treated as the multiword lexemes, but mostly there are not present in plWordNet and, moreover, the elimination of such occurrences of training words would decrease the limited number of examples for some sense. That is why we decided to annotate training words in collocation occurrences with their literal meaning in some cases. In the example above, *język* was annotated with the sense *tongue* (i.e. body part).

We had started with annotating all training word occurrences in the documents of the subcorpus, but shortly we realised that our time limitations are too tight and we would get a very imbalanced numbers of training examples for subsequent senses, as the sense frequencies vary a lot. So, in the second phase of manual annotation we tried to identify only examples including less frequent senses[1]. The whole subcorpus includes about 1 500 semantically annotated training word occurrences, however, the sense frequencies are not still balanced yet. For some senses we could find only few examples in the whole IPIC. The detailed numbers concerning annotated occurrences of training words and their senses are presented in Table I.

TABLE I
ANNOTATED TRAINING WORDS AND SENSES.

| Word | No. of senses | Annotated senses | No. of examples |
|---|---|---|---|
| *agent* | 4 | 69/22/2/123 | 216 |
| *automat* | 3 | 28/31/46 | 105 |
| *dziób* | 4 | 31/17/24/9 | 81 |
| *język* | 2 | 22/54 | 76 |
| *klasa* | 6 | 9/51/6/13/29/7 | 115 |
| *linia* | 6 | 15/4/35/12/4/12 | 82 |
| *pole* | 3 | 69/25/2 | 96 |
| *policja* | 2 | 23/41 | 64 |
| *powód* | 2 | 137/122 | 259 |
| *sztuka* | 6 | 5/12/37/19/12/11 | 96 |
| *zamek* | 4 | 19/36/19/18 | 92 |
| *zbiór* | 7 | 16/16/1/15/3/32/3 | 86 |
| *zespół* | 5 | 60/1/3/28/29 | 121 |

## III. APPLIED WSD ALGORITHMS

We based our current work on the previous experience from the creation of WSD systems, which were constructed mainly for English and were described in literature, e.g. [8]. We wanted to investigate behaviour of several known approaches adapted to the Polish language and Polish language tools and resources. We assumed the following scheme of processing:

1) Morphosyntactic processing of the training corpus (there is no shallow parser available for Polish).
2) Extraction of feature vectors describing occurrences of training words and storing them in the ARFF format (*Attribute-Relation File Format*).
3) Training and testing classifiers in the *Weka* system [11].

The key issue is the choice of types of features that will be used in training vectors. We surveyed types of features most frequently used in WSD systems, e.g. [8]. Lacking more advanced language tools, we assumed as the basic paradigm the *bag of words* model—Yarowsky and Florian [12] showed that omission of bag of words resulted in the decreased accuracy. Finally we have chosen five types of features:

1) *Parts of Speech* in the ±2 text window around the training word occurrence (PoS)—information concerning parts of speech, or more precisely more fine grained division into 32 *grammatical classes*, comes from the TaKIPI morphosyntactic tagger [13] applied during preprocessing of the corpus. A training vector includes numerical identifiers of grammatical classes.

[1] A similar strategy was applied during the construction of the Basque semantically annotated corpus built for the Senseval-3 competition [10]. In this corpus, the minimal number of training examples per word $w$ was set to $N_w = 75 + 15 \times |senses(w)|$, where $senses(w)$ is the set of senses of $w$.

2) *Collocation words* in the $\pm 2$ text window (Coll)—word base forms of all grammatical classes (except punctuation signs) which occurred in the close context of the $\pm 2$ text window. The lematisation was done by TaKIPI. No statistical filter was applied to the found occurrences. Found word base forms are represented by identifiers in the training vector.
3) The first noun to the left and to the right of the training word occurrence (Nouns)—nouns are important meaning bearers in the text. Information concerning nouns with which the training words is associated in text can be an important factor in determining the sense of the training word. Nouns are analysed on the level of their base forms. The appropriate elements of the training vector store the numerical identifiers.
4) *Wider context* of the training word (henceforth WCont)—described by base forms of words occurring in a larger text window. In a way typical for the bag of words model, the context is represented by the boolean vector, where 1 means that the corresponding base forms occurred in the given context. For the selection of base forms for the representation of contexts, we tested three methods:
   - selection by frequency—only most frequent base forms,
   - method of Ng and Lee [14],
   - selection by the Quantity of Information [15].

Concerning the description of the larger context, the selection of the most frequent base forms (4) is the simplest one. All base forms occurring in the context of training words more than the established threshold $k$ are included in the set describing the context—the bag of words. The value of $k$ is set experimentally.

According to the Ng and Lee method (4) we try to estimate the probability $p(s|b)$ of describing the given sense $s$ of the training word $w$ by the given base form $b$ occurring in the context of $w$ [14]:

$$p(s|b) = \frac{f(s,w,b)}{f(w,b)} \times \frac{1}{f(b)} \tag{1}$$

where:

- $f(s,w,b)$ is the frequency of $b$ in those contexts in which $w$ occurs in the sense $s$,
- $f(w,b)$—the frequency of $b$ in the contexts of $w$ in any sense,
- $f(b)$—the total frequency of $b$ in the whole corpus.

In the measure of the Quantity of Information we attempt calculating how characteristic is the given base form $b$ for the sense $s$ of the training word $w$, i.e. how much information it delivers [15]:

$$Q(b,w) = -log\frac{1+N(b,w)}{1+|senses(w)|} \tag{2}$$

where $N(b,w)$ is the number of senses of $w$, such that they co-occur with $b$ in the context.

We can also filter base forms used in the context description by a stop list or by the morphosyntactic properties of their occurrences, e.g. filtering out base forms of some grammatical classes.

## IV. Experiments

For the experiments we concentrated our attention on Machine Learning algorithms which are implemented in Weka 3.4.14 [11] and which can be applied to small sets of training examples. Finally, we selected three algorithms representing different types of classifier used for WSD:

- *Naïve Bayes* (henceforth NB)— representing probabilistic methods in WSD,
- *k Nearest Neighbours* (kNN)—methods based on similarity to examples,
- *Decision Tables* (DT)—methods based on discriminating rules,

All experiments were done in *Weka* environment on the basis of previously prepared vectors describing training examples. Evaluation was performed in *Weka Experiment Environment*. Because of the small size of dataset, we used the scheme of *leave-one-out cross-validation* for all tests.

### A. Thresholds selection

In the first set of experiments we tried to discover sub-optimal values for the subsequent thresholds used in the methods of base form selection for the wider context. In the case of the Ng and Lee method and Quantity of Information we were looking for the values of both factors that result in higher WSD accuracy. For selection by frequency we were looking for the minimal frequency threshold above which the base forms have a positive influence on the accuracy.

In the case of all selection methods the higher the threshold is the smaller is the number of base forms used for the description of contexts. We remove less useful descriptors—base forms—and decrease the level of noise in data by increasing the thresholds values. As different Machine Learning algorithm (and constructed classifiers) express different possibilities in coping with noise in training data, we had to define separate sets of threshold value for the subsequent classifiers.

The used selection methods can be divided into two groups. The Ng and Lee method takes into account frequencies of base forms collected for the whole corpus. On this basis it eliminates base forms that do not have influence on the WSD process, like conjunctions and prepositions. This method has problems only with elimination of numerals, as lexemes not contributing to WSD.

The two other methods, i.e. Quantity of Information and selection by frequency do not include a similar mechanism. Thus, we had to extend them with a manually created filtering rules and a stop-list to eliminate such informationally vague base forms[2].

[2] Conjunctions, prepositions and numerals comprise about 18% of tokens in IPIC [16].

TABLE II
THRESHOLD VALUES ESTABLISHED EXPERIMENTALLY FOR THE CONTEXT DESCRIPTION SELECTION.

| | Method of selection | | |
|---|---|---|---|
| Classifier | Frequency | Ng and Lee | QI |
| NB | >2 | >0.001 | >0.5 |
| kNN | >6 | >0.001 | >0.7 |
| DT | >4 | >0.001 | >0.2 |

TABLE III
COMPARISON OF AVERAGE ACCURACY ACHIEVED USING DIFFERENT CLASSIFIERS AND SELECTION METHODS ON THE REDUCED SET OF TRAINING WORDS.

| | Classifier | | |
|---|---|---|---|
| Method | NB[%] | kNN [%] | DT[%] |
| Frequency | 75.92 | 57.74 | 67.92 |
| Ng and Lee | 79.02 | 74.33 | 66.54 |
| QI | 78.54 | 69.10 | 69.50 |

The first experiments were performed on the data collected for four selected training words: *agent*, *sztuka*, *dziób* and *zamek*. The reason for this limitation was the high computation cost of the experiments. As the result we took the average from all experiments. During the experiments the training vectors included only the wider context features (WCont), as only these features are influenced by the selection methods.

The final sub-optimal values for thresholds were identified on the basis of analysis of values obtained for all base forms occurring in the contexts of the four training words. The obtained threshold values are presented in Table II—they were consequently applied in all following experiments.

In the case of the selection by frequency, results are dependent on the size of the text window and the number of training examples. The size of the window was set to ±20 tokens. After changing the size we would have to define the threshold value again.

The highest threshold values were obtained for the KNN algorithm for all three methods. Table II shows that it is more sensitive to noise in data in comparison to the other two methods, which was expected. However, limiting the context description to the most informative base forms can increase its accuracy to a large extent.

The results obtained for the Ng and Lee method are similar to the results achieved for English [14].

Having the sub-optimal threshold values extracted, we compared all three methods of selection on the data set of the four training words. The results of the comparison presented in relation to all three classifier types are given in Table III.

In the case of both: Naïve Bayes and kNN the best results were achieved while using the Ng and Lee method. Contrary, Decision Table produced the best result in combination with the selection based on Quantity of Information.

### B. Feature selection

In the next set of experiments, we wanted to identify a sub-optimal set of features for the description of training data. We performed these test on the full set of 13 training words. As all training words are nouns we could omit feature expressing grammatical class of the word being disambiguated (in the centre of the context) without loss of information.

TABLE IV
AVERAGE ACCURACY FOR ALL TRAINING WORDS IN RELATION DIFFERENT SETS OF TRAINING FEATURES.

| | Classifier | | |
|---|---|---|---|
| Features used | NB[%] | kNN [%] | DT [%] |
| WCont | 89.80 | 72.94 | 75.61 |
| WCont + PoS (±2) | 88.79 | 70.08 | 77.88 |
| WCont + Coll(±2) | 88.88 | 71.24 | 73.52 |
| WCont + Nouns | 77.83 | 62.51 | 66.25 |

As the starting point we assumed the wider context set of features (one feature for each base form included to the context description). Next we extended WCont with combination of the other types of features. The results being average from all 13 training words are presented in Table IV.

On the basis of the average results for all words (Table IV) we can observe that only Decision Table classifiers produce higher result after adding PoS (±2) features to the training vectors. The detailed analysis of the results obtained for subsequent words showed that the influence of PoS (±2) features varies significantly in relation to particular words. For example, in the case of the training word *policja* and the kNN classifier after adding the PoS (±2) features its accuracy increased by 7.8%, while in contrast, for the same word and Decision Tables, the combination of the WCont features with the Coll(±2) features increased the accuracy by 34.4%. Such differences can be explained on the basis of the inspection of training examples. There are two senses distinguished for the word *policja* in the corpus:

- (English *police*) an institution protecting order and safety,
- (English *police station*) a place—a police station.

In the case of the second sense, the word *policja* is very often a part of the adverbial place describing some placement or destination, e.g. *pojechał na policję* (*went to the police station*). This kind of association is barely visible according to WCont features (bag of words), but after adding collocation words or grammatical classes from the nearest context as features becomes much more prominent.

We noticed that Naïve Bayes and kNN classifiers react in a similar way to the extension of training vectors by additional features. In contrast, the Decision Table classifier returns identical results for many words regardless of changes in the training features set. It can be caused by the low values achieved by this features and the lack of their influence on the final decisions.

### C. Analysis of the results

The best results achieved for subsequent words using different classifiers are presented in Table V. The accuracy was calculated in all cases according to the one-leave cross-validation, and only the parameters of classifiers varied.

Base line has been calculated as the ratio of the number of examples for the dominating sense of the given word to the total number of examples for the given word, i.e. the base

TABLE V
BEST RESULTS IN ONE-LEAVE CROSS-VALIDATION FOR SUBSEQUENT TRAINING WORDS IN RELATION TO THE USED CLASSIFIERS.

| | Classifier | | | |
|---|---|---|---|---|
| Word | NB [%] | kNN [%] | DT [%] | base line |
| *agent* | 93.98 | 88.43 | 94.44 | 56.94 |
| *automat* | 90.95 | 64.76 | 57.14 | 43.81 |
| *dziób* | 76.54 | 64.2 | 59.26 | 38.27 |
| *język* | 78.95 | 71.05 | 71.05 | 71.05 |
| *klasa* | 82.5 | 56.67 | 63.04 | 44.35 |
| *linia* | 54.88 | 47.56 | 47.56 | 42.68 |
| *pole* | 95.83 | 87.5 | 91.67 | 71.88 |
| *policja* | 87.5 | 79.69 | 98.44 | 64.06 |
| *powód* | 84.17 | 74.9 | 91.12 | 52.90 |
| *sztuka* | 57.29 | 53.13 | 59.38 | 38.54 |
| *zamek* | 86.96 | 79.35 | 66.3 | 39.13 |
| *zbiór* | 79.07 | 50 | 60.47 | 37.21 |
| *zespół* | 74.38 | 67.77 | 67.77 | 49.59 |
| **average** | 80.23 | 68.08 | 71.36 | 50.03 |

line equals the accuracy of a simple majority classifier. For all words, base line calculated in this way is much higher than the base line of a random choice (compare Table I for the number of senses).

Almost all results are higher than the base line. Only for the word *język* the kNN and Decision Table classifiers produced results comparable to the base line.

One can notice the worst results, i.e. being only slightly above the base line, were achieved for words: *język*, *linia* and *sztuka*. In the case of *linia* and *sztuka* one could expect such resuts, as both words posses several polysemous senses, which are difficult to be differentiated. The low results for the word *język* is a little surprising, and it is probably an artefact of the training corpus.

The best results were achieved for the words: *zamek* and *agent*. All sense of the first one are exactly homonyms, so the meaning differences should be clear for classifiers. On contrary, the good result of *agent* is biased by the training corpus to some extent. The examples for two from the four senses of *agent* come from the same set of documents of the very similar type: the set of legal documents produced in the Parliament of Poland (*Dziennik Ustaw*). Thus the characteristic vocabulary occurring in these documents could simplify the differentiation of these two senses and could increase the average score.

Moreover, one should remember, that all the words being disambiguated are nouns, and for nouns the results achieved in WSD are mostly higher than for other Parts of Speech.

Large part of the results supports our initial assumptions that words with many homonymous senses are easier for WSD. However, we could also observe some exceptions to this scheme, which can be explained on the basis of detailed analysis of the results and the training corpus.

## V. CONCLUSIONS

During the performed experiments, WSD algorithm based on the Naïve Bayes achieved the accuracy 30% above the baseline on average, the Decision Table classifier 21% above the base line on average, and the kNN classifier 18% above the base line. It is worth to emphasise that the worst results was produced by the classifiers based on kNN algorithm, which is claimed in literature, e.g. [8], to be one of the best for WSD. But according to Escuedro [13], we need to introduce weighting of examples and features and sophisticated similarity metrics in order to achieve better accuracy with kNN than with Naïve Bayes. As we wanted only to compare different approaches in relation to Polish, we did not apply such extensions. For the kNN algorithm, the introduction of the different values for the $k$ parameters could be helpful, as well.

We got also quite low results while extending training vectors with additional features, that is often performed in WSD systems. But a closer look into the detailed results for subsequent words shows that in the case of at least some words (*policja*, *język*, *powód*, *automat*) the accuracy increased with additional features. It seems that the optimal solution is selection of different sets of training features for subsequent words and applying them in relation to a word being disambiguated. It shows that the general schemes worked out for English should not be directly transferred to Polish.

The most informative type of features appeared to be the wider context, i.e. the bag of words approach. Its positive influence can be even increased by the application of a thesaurus and grouping synonyms or clustering based on automatically extracted Measures of Semantic Relatedness [17], [18], [19].

The main disadvantage of the wider context based representation is its strong dependence on the type of text for which it is applied. It is especially visible in the case of specific texts like: legal texts or scientific works, in which a specialised vocabulary is over-represented[3]. Words from this specific vocabulary rarely occurs in the rest of a large corpus, so are highly ranked by the automatic methods of selection. Classifiers trained on the basis of the wider context representation are difficult to transfer from one domain to the other.

Moreover, we noticed that the wider context representation is especially sensitive to some errors in the corpus annotation. For example, in the first phase of corpus annotation all occurrences of training words were annotated. In the case of some occurrences, training words were located very close to each other even in the range of the text window of the wider context. So the same base form occurrences were taken to the representation of more than one training word, e.g. we present below a snippet from the training corpus which includes two occurrences of the word *automat*:

"*Nikt natomiast nie ubezpieczył się od zabicia przez automat z zimnymi napojami. A takie automaty zatłukły już 15 osób, które usiłowały siłą wyciągnąć puszkę po wrzuceniu pieniędzy i bezskutecznym naciskaniu guzika [...]*"

(*Nobody has insured himself from being killed by a drinks machine. In the meantime such machines have already beaten to death 15 persons who were trying to pull out a can using*

[3] However, training classifier based on the wider context representation on a representative subcorpus is its advantage.

*force after they had thrown money into it and had been pressing a button without result.*)

In the example above, the distance between the occurrences of *automat* is 6 tokens, i.e. much less than the text window of the wider context. The descriptions of both occurrences are collect from the same tokens in large extent. Thus two very similar training examples are constructed. During training they can introduce some bias, when the number of training examples is small, during testing they can increase the result while separated into the training and testing part of the corpus. In order to avoid such cases, we need to carefully select locations of training examples or to filter an existing corpus.

The results of our investigations are very positive for processing the Polish language. The performed experiments showed the construction of WSD system for Polish on the basis of limited language resources and tools is possible. Obviously with the larger number of disambiguated words one can expect decrease in the average result, but still methods developed for English seem to work for a typologically different language like Polish.

## Acknowledgment

Work financed by the Polish Ministry of Education and Science, project No. 3 T11C 018 29.

## References

[1] Y. Wilks, “Preference semantics,” in *Formal Semantics of Natural Language*, E. L. Keenan, Ed. Cambridge, UK: Cambridge University Press., 1975, vol. III, pp. 329–348.

[2] B. L.-T. Adam Przepiórkowski, Rafał L. Górski and M. Łaziński, “Towards the national corpus of polish,” in *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08)*, E. L. R. A. (ELRA), Ed., Marrakech, Morocco, may 2008.

[3] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda, “Words, concepts and relations in the construction of Polish WordNet,” in *Proc. Global WordNet Conference, Seged, Hungary January 22-25 2008*, A. Tanács, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen, Eds. University of Szeged, 2008, pp. 162–177.

[4] P. Edmonds, “Introduction to senseval,” *ELRANewsletter*, vol. October 2002, 2002.

[5] B. Broda and M. Piasecki, “Experiments in documents clustering for the automatic acquisition of lexical semantic networks for polish,” in *Proceedings of the 16th International Conference Intelligent Information Systems*, 2008, to appear.

[6] B. Broda, M. Piasecki, and S. Szpakowicz, “Sense-based clustering of polish nouns in extracting semantic relatedness,” June 2008, to appear in the AAIA’08 Conference Proceedings.

[7] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science PAS, 2004.

[8] E. Agirre and P. Edmonds, Eds., *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

[9] M. Piasecki, G. Godlewski, and J. Pejcz, “Corpus of medical texts and tools,” in *Proceedings of Medical Informatics and Technologies 2006*. Silesian University of Technology, 2006, pp. 281–286.

[10] T. Pedersen, “The duluth lexical sample systems in Senseval-3,” in *Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona*, 2004, pp. 203–208.

[11] Weka, “Weka 3: Data Mining Software in Java,” 2008, http://www.cs.waikato.ac.nz/ml/weka/.

[12] D. Yarkowsky and F. R., “Evaluating sense disambiguation across diverse parameter spaces,” *Journal of Natural Language Engineering*, vol. 8, no. 4, pp. 293–310, 2002.

[13] M. Piasecki, “Polish tagger TaKIPI: Rule based construction and optimisation,” *Task Quarterly*, vol. 11, no. 1–2, pp. 151–167, 2007.

[14] T. Ng and H. Lee, “Integrating multiple knowledge sources to disambiguate word senses: An examplar-based approach,” in *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, 1996, pp. 40–47.

[15] C. Loupy, M. El-Béze, and P. Marteau, “WSD based on three short context methods.” in *SENSEVAL Workshop, Herstmonceux Castle, England*, 1998.

[16] A. Przepiórkowski, “The potential of the IPI PAN Corpus,” *Poznań Studies in Contemporary Linguistics*, vol. 41, pp. 31–48, 2006. [Online]. Available: http://nlp.ipipan.waw.pl/~adamp/Papers/2005-psicl-numbers/

[17] M. Piasecki, S. Szpakowicz, and B. Broda, “Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns,” in *Proc. Text, Speech and Dialog 2007 Conference*, ser. LNAI, vol. 4629. Springer, 2007.

[18] M. Piasecki, S. Szpakowicz, and B. B., “Extended similarity test for the evaluation of semantic similarity functions,” in *Proc. 3rd Language and Technology Conference, October 5-7, 2007, Poznań, Poland*, Z. Vetulani, Ed. Poznań: Wydawnictwo Poznańskie Sp. z o.o., 2007, pp. 104–108.

[19] B. Broda, M. Derwojedowa, M. Piasecki, and S. Szpakowicz, “Corpus-based semantic relatedness for the construction of polish wordnet,” in *Proceedings of the 6th Language Resources and Evaluation Conference (LREC’08)*, 2008, to appear.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 79 – 82

ISBN 978-83-60810-14-9
ISSN 1896-7094

# USERING. Educational Self-Tuning–Recommendations in the 8th Level of ISO/OSI Model

Miroslaw Bedzak
Institute of Control Engineering
Szczecin University
of Technology,
ul. Sikorskiego 37,
Poland
Email: bedzak@ps.pl

***Abstract*—It is autumn 2012…The VMware Infrastructure …3, 4 editions virtualised crucial components of IT environment, i.e. computing (CPU, RAM), networking and storage. However, an important element was overlooked. Which one? A user. There was no mechanism built in into VI3/VI4 that would support administrator in gaining effectively the skills of implementing solutions advised by manufacturer, the so-called recommendations (“best practice”, etc.). Either, the high level of user’s skills (primarily, administrator’s ones) were not treated as a valuable resource of LAN infrastructure that could be (should be) used, while a “surplus” could be virtualised for a common good of the society (local and/or global), concentrated around the enterprise-class infrastructure virtualisation technology. The latest edition, VI5 beta, brings also an important change in this respect in the form of a new module, VMware Usering, which is directed in the current version (i.e. beta version) first of all towards security hardening, i.e. the linguistic inference has been accomplished by a method of fuzzy control basing on the knowledge-base built on the recommendation of VI5 beta manufacturer. The VMware Usering is a set of tools, stimulating users to take up actions being consistent with manufacturer’s recommendations (VMware User Hardening) and virtualising (optionally for administrators with HIGH $_{\text{FUZZY}}$ rating) the competence resource of advanced users (VMware User Competence Sharing). The rise of a resistance level of IT infrastructure to increasing threats of internet security is also a manufacturer’s own interest, therefore one can expect in the near future a popularisation of VMware Usering-class solutions as an important tool supporting an average internet surfer in his/her solitary struggle against crackers at the level of 8th OSI model layer .**

## I. Introduction

IT IS autumn 2015… Computer software is not only a code with specific functionality. It is also – and perhaps first of all – a knowledge, experience and competences of computer programmers enclosed (also outside the code) in a set of the so-called recommendations. Owing to them, a user can avoid obstacles and traps of the interference of tens (hundreds) of options (chances) “at choice”. Obviously, a user can but does not have to… The tandem of software functionality and user competence (including the use of recommendations) determines at last the quality of IT solution.

Apart from controlling and managing a specific functionality of IT system, an important issue (in present-day internet times) is to pay due attention (at least 10% of daily work time with IT system?) to IT security questions. Besides implementing hardware and software solutions, a key element is the quality of the weakest link, i.e. the quality of knowledge and user competence.

Probably the IT infrastructure that works on-line with the Internet is helpless (except for entries in system logs) when administrator has turned off (perhaps unintentionally) its protecting systems, i.e. firewall/IDS/IPS (exposing the same its resources to the prey of crackers). With some exaggeration, we can probably describe the past IT systems as “deaf-mute” ones, i.e. helpless on the one hand “to admin insanity/ignorance”, while not using completely the experience and professionalism of brilliant user (administrator) on the other hand. There was no feedback in them: “reprimand for making and sticking to an error” and first of all rewarding “good behaviour consistent with manufacture’s recommendations” [1]-[3].

Two extremely popular previous versions, VI3 and VI4, virtualised crucial hardware and software components of IT infrastructure, forgetting however about a key link (for its proper functioning), i.e. about a user.

Together with VI5 beta Enterprise, we are receiving a tool, VMware Usering, stimulating (supporting intensively) a user for taking up actions that are consistent with manufacturer’s recommendations, i.e. VMware User Hardening (VMware UH) as well as a tool virtualising the competence resource (of advanced users with f(UR):= HIGH rating), i.e. VMware User Competence Sharing (VMware UCS).

## II. Usering Security, or Educational Self —Tuning in the 8th Layer of OSI Model.

### *A. Admin in the “open loop” of acquiring competence/ knowledge/ experience.*

The detailed discussion of VMware Usering mechanism surpasses the frames of the present paper. It is sure enough that descriptions referring to most new mechanisms available in VI5 edition will show in the near future. Below, only the essence of VMware UH and VMware UCS operation will be

978-83-60810-14-9/08/$25.00 © 2008 IEEE

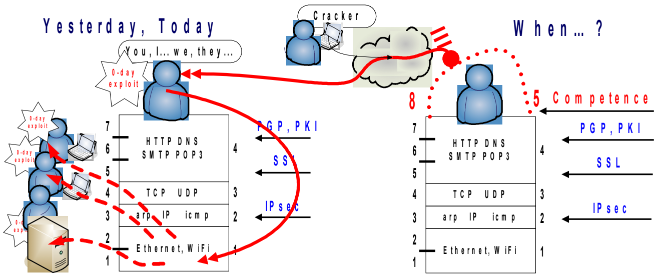


Fig 1 . User Hardening, a tool supporting the user in counteracting internet attacks from the 8th layer of OSI model: Competence layer.

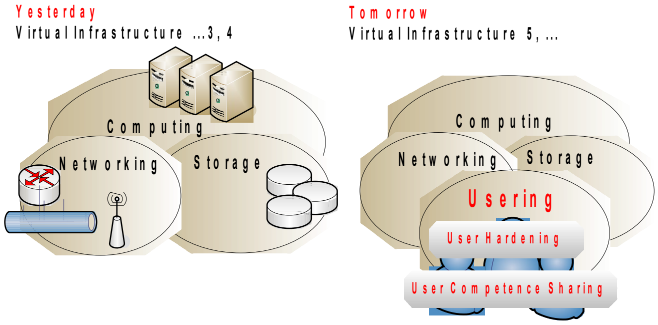


Fig 2. Usering, next virtualized component of the enterprise-class IT environment.

illustrated on selected examples. Not many users have knowledge of availability of VMware UH also for older editions of Virtual Infrastructure 3 and 4 (after installing additionally manufacture's patches to Virtual Center server ver. 2 or 3). In the present version (i.e. VI5 beta), VMware UH is using manufacturer's recommendations that are connected with IT security hardening, i.e. Vmware UH.

Until recently, almost all of us have accepted quietly a common practice that responsibility of admin is to "become" (of a sudden the best) or "becoming" an expert (at last after many months /sometimes many years) (No 1 problem), who will be able, apart from managing classic IT resources, to "keep a tight rein on" (meaning: extend knowledge of) a common user of virtual infrastructure.

The no 1 problem is a process (frequently long-lasting one) of coming in admin competence to the knowledge level of software engineers (authors) (VMware team): from studying "case study"/"best practice" "helps" /pdf files/ through specialist courses to own experience/experimenting with test/ production network. Direct contact/exchange of experience (meaning: sharing with each other) between computer programmer/software engineer (author) and a user/administrator has been out of question (apart from a small group using Help Line). Out of hundreds / thousands of documentation pages, few users "shell out" most important procedures, primarily those advised by manufacturer, i.e. recommendations. And what is the effect of this?

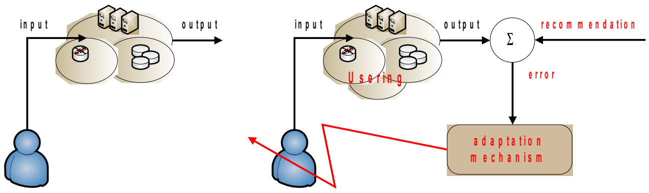


Fig 3. User in the "opened (left)/ closed (right) loop", or educational self-tuning.

The IT system has quietly permitted sometimes to "demolish" itself, or sometimes to "tune up perfectly", depending on admin competence. A common practice, used also by a manufacturer, was to record important events (unfortunately) in many scattered logs/data-bases. Every admin will admit that effective daily tracking (and first of all correlating any number) of tens/hundreds/thousands of scattered events is almost impossible with the quality comparable at least to solutions of Intrusion Detection/Prevention System type. The process described above can be summarised unfortunately as "admin all alone in the open loop of acquiring knowledge":

- lack of built-in mechanism that evaluates *on-line* and *on-time* the conformity of user actions (including admin) with the advised manufacturer's recommendations, e.g. admin can of course do "everything, but it will be good if he/she does not take up actions that decrease the system security, does it?
- lack of mechanism (rewarding/promoting one) that uses user competence and acquired knowledge for the good of local/global user society concentrated around a specific technology (except for enterprise discussion lists),
- similarly to counteracting the waste of resources of the non-virtualised server computing type, we all agree with the thesis that one can counteract with equal determination the waste of acquired knowledge and competence of professional users.

### B. Admin in the "closed loop", or educational self-tuning.

When closing feedback loops, we receive immediately profits:

- administrator sees "without delay" the effect of his/ her actions (choices of options, configurations, activation/deactivation…, etc.) on "educational self-tuning error" (minimum one the best) with respect to advised manufacturer's recommendations,
- evaluation of "educational self-tuning error" (difference between user choice/decision and manufacturer's recommendation) is made (in the present version, i.e. VI5 beta) with fuzzy logic method: except for interference of the "black-white" type, i.e. zero/"no conformity"/"maximum error" vs. one/"100% conformity"/"zero-value error", we are using affiliation degrees (set of real numbers within a range of ≤0; 1 ≥) for a fuzzy set that represents one of the values of linguistic variable *competence* { *Low, Medium, High* }:

error of educational self-tuning $_{FUZZY}$ := recommendations of manufacturer $_{FUZZY}$ – decisions of administrator $_{FUZZY}$

- apart from the c onformity with important manufacturer’s recommendations, one can evaluate “educational self-tuning error” for a common user who, when acting within authorisations (permissions), i.e. making use of trappings (privilege) attributed to his/her part (role), can be “rewarded with a neutral mark or higher *f* ( *UH* )” for “moving” inside the cage determined by the role” and “punished with a neutral mark or lower *f* ( *UH* )” even for taking up “only attempts” to make use of privileges not attributed to his/her role”.

### C. *VMware User Hardening $_{security}$ (VMware UH $_{security}$ ).*

Abstract he mechanism that allows for closing the educational loop of feedback is just VMware UH module (apart from VMotion, VMware HA, VMware DRS and two other novelties in VI5 beta version, it is the next option for Standard and Enterprise versions requiring the licence). In VMware UH, the interference has been accomplished by a method of fuzzy control (computational intelligence), basing on the knowledge-base constructed on manufacturer’s recommendations [6],[7].

UH $_{security}$ : User Hardening $_{security}$

$e_{FUZZY} \cong r_{FUZZY} - d_{FUZZY}$

- ○ e {$e_1$ , $e_2$ … $e_n$ }- error of educational self-tuning
- ○ r {$r_1$ , $r_2$ … $r_n$ } - recommendations of manufacturer
- ○ d {$d_1$ , $d_2$ … $d_n$ } - decisions of administrator
- *f* ( *UH* ):= { *LOW* $_{FUZZY}$ , *MEDIUM* $_{FUZZY}$ , *HIGH* $_{FUZZY}$ }
- Objective: local evaluation (automatically and without delay) of administrator’s (user’s) implementation of recommendations (of manufacturer)

The key element of fuzzy control is expert knowledge-base constructed on manufacturer’s recommendations:

(+) It is recommended…

(-) VMware does not recommend using…

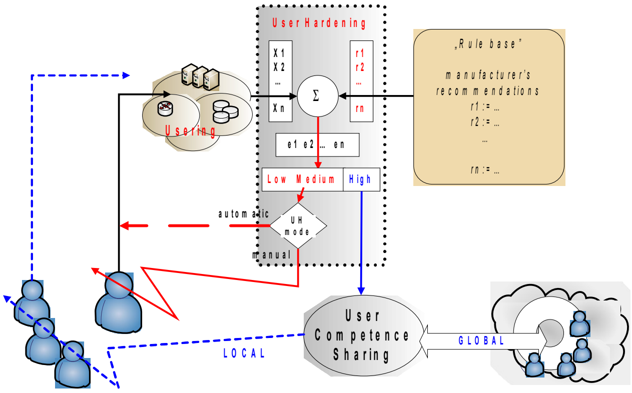


Fig 4. Concept of using computational intelligence in the “closed loop” of educational self-tuning of User Hardening module-tuning.

written in the form of (IF… Then…) rules

IF *premiss1* AND *premiss2* AND…

Then *conclusion1* AND *conclusion2* …

that use the values of linguistic variable fuzzy logic described by means of appropriate fuzzy sets. The VMware UH does not analyse “every step” of user; it take into account only these actions, for which a manufacturer’s recommendation exists in the knowledge-base. In the supplement Appendix, the examples of manufacturer’s recommendations are given (scattered in rich documentation) that increase the IT security of infrastructure with respect to its crucial components: ESX Server Host, Service Console, Virtual Machine and VirtualCenter.

In order to illustrate the essence of VMware UH $_{security}$ operation, we will use the first recommendation from the Appendix that refers to ESX installation and evaluate a possible “educational self-tuning error”, taking into account:

- default setting of manufacturer,
- decisions of administrator,
- recommendations of manufacturer,

hat is:

- error of educational self-tuning $_{FUZZY}$ := recommendations of manufacturer $_{FUZZY}$ -
  – decisions of administrator $_{FUZZY}$

*f* ( *UH* ):= *HIGH* $_{FUZZY}$

What does it mean in reality? As early as a few minutes of ESX 3 installation, administrator should “brake down” default settings (but not recommended by VMware!) in case of production environment (and not a test one) in order to ensure HIGHER security and aim at *f* ( *UH* ):= *HIGH* $_{FUZZY}$ mark [4], [5] :

- default setting “Create a default network for virtual machines” is not recommended for production environment by VMware manufacturer.

„ *[…] If the “Create a default network for virtual machines” is selected, virtual machine network traffic will share this adapter with the service console. This is not a recommended configuration for security purposes”*

Conclusion:

- as early as a few minutes of administrator contact with infrastructure software, we can evaluate a probable

- error of educational self-tuning $_{FUZZY}$ :=

recommendations of manufacturer $_{FUZZY}$ – decisions of administrator $_{FUZZY}$

and interfere according to fuzzy control nomenclature about evaluating admin competence of the type:

- increase
- neutral
- decrease

receiving, e.g.:

$\mu$ ( LOW FUZZY ):= 0.0; $\mu$ ( MEDIUM FUZZY ):= 0.2;

$\mu$ (HIGH FUZZ ):= 0.8 $\Rightarrow$ :→ (UH):= 82% HIGH FUZZY

### D. *VMware UH $_{security}$ : manual – automatic (autopilot) mode.*

The VMware UH $_{security}$ (in VI5 beta version) default operation mode is *manual* , i.e. presenting the error of educational self-tuning (with indication to recommended options) without enforcing choices/decisions/methods of user action consistent with manufacturer’s recommendations. An interesting mode is *automatic* , which for the profile of VMware UH $_{security}$ edition, as a *security hardening* , will approve (similarly to the *transaction* mechanism in the data-base nomenclature) only these decisions of user, which will not decrease (but rather increase) the global level of infrastructure security for production network.

## III. VMware User Comptetence Sharing (UCS)

Natural consequence of minimising the error of educational self-tuning $_{FUZZY}$ of the competence resource for a specific user of VMware UH module at a respectively high level (e.g. over 75%) HIGH $_{FUZZY}$ is to move to next stage on the way to full Usering, i.e. to virtualise the competence resource of advanced users with VMware UCS module by "sharing"/"participation" for the good of local and/or global user society. The idea of VMware UCS functioning is springing from the mechanism of self-education of professional internet discussion list users. A particularly good example is the mechanism that supports evaluation of the importance/quality of post contents of the moderated discussion list VMWare VMTN Discussion Forums...Novice, Expert, Champion, and Guru. The VMware UH $_{\text{security}}$ fulfills a similar role to a moderator and opinions of internet surfers, but at a local level in evaluating the practical competence of a user who manages the advanced infrastructure.

VMware UCS uses of course a VMware UH $_{\text{security}}$ filter :

VMware UH $_{\text{security}}$ : *filter_HIGH* {user1_ *LOW* , user2_ *HIGH* , … userN_ *MEDIUM* }= user2 and can work in the following modes (different possibilities are tested in the present beta version; final VMware UCS operation modes should be determined within the nearest months after all test are concluded):

- Local_info: users indicated by user2 (all, selected,…) will be familiarised with the current ranking of VMware UH $_{\text{security}}$ evaluation,
- Local_sharing: during taking up actions that are inconsistent with manufacturer's recommendations, a user has to receive a counter-signature from the user with HIGH mark,
- Global _sharing: a user2 user with HIGH mark receives a possibility/invitation from VMware manufacturer for co-operation for the good of the society concentrated around the product (its scope and form is determined by manufacturer).

## IV. Conclusion

Internet surfers should be equipped with knowledge and competence that allow on the one hand for effective acquiring of advanced skills for daily management of (work with) modern IT systems, while enable counteracting against misuses from the part of less or more organised groups of internet crackers on the other hand. Because quick achievement of the aforesaid objectives is important for the well-comprehended business-like own interest of manufacturer, thus we can expect in the near future a popularisation of solutions of the VMware Usering class as an important tool that supports, among others, an average internet surfer in his/her alone struggle with crackers at a level of the 8th layer of OSI model. Undoubtedly, the strength of this solution lies in the skilful connection of a tool set that stimulates users for taking up actions, which are consistent with recommendations of VMware UH manufacturer and (optionally for administrators with $f(UH)$:= $HIGH_{FUZZY}$ mark) virtualising the competence resource of VMware UCS users.

## Appendix

Examples of manufacturer's recommendations that increase IT security of infrastructure with respect to its crucial components [8]-[11].

### ESX Server Host:

Do Not Create a Default Port Group

Use a Dedicated, Isolated Network for VMotion and iSCSI

VMware best practices recommend that the service console and VMotion have their own networks for security reasons

Do Not Use Promiscuous Mode on Network Interfaces

Protect against MAC Address Spoofing (MAC address changes, Forged transmissions)

Secure the ESX Server Console

Mask and Zone SAN Resources Appropriately

Protect against the Root File System Filling Up

### VirtualCenter:

Manually changing Most Recently Used to Fixed is not recommended. The system sets this policy for those arrays that require it. For active/passive storage devices, Most Recently Used is highly recommended

Comparing Raw Device Mapping to Other Means of SCSI Device

Virtual Machine

Disable Unnecessary or Superfluous Functions

Limit Data Flow from the Virtual Machine to the ESX Server Host

Isolate Virtual Machine Networks

Minimize use of the VI Console

### Service Console:

Isolate the Management Network

Configure the Firewall for Maximum Security

Use VI Client and VirtualCenter to Administer the Hosts Instead of Service Console

Use a Directory Service for Authentication

Strictly Control Root Privileges

Limiting Access to su. Using sudo

Establish a Password Policy for Local User Accounts

Limit the Software and Services Running in the Service Console

Do Not Manage the Service Console as a Linux Host

Establish and Maintain File System Integrity

Maintain Proper Logging

## References

[1] R. J. Anderson, "Security Engineering: A guide to building dependable distributed systems," 2001, Wiley & Sons.

[2] N. Ferguson and B. Schneier, "Practical Cryptography," Wiley & Sons, 2003.

[3] N. F. Johnson and S. Sushil Jajodia, "Steganography: Seeing the Unseen," IEEE Computer, 1998, February, 26-34.

[4] C. Chaubal: Security Design of the VMware Infrastructure 3 Architecture, VMware, 2007.

[5] C. Chaubal: VMware Infrastructure 3. Security Hardening, VMware, 2007.

[6] L. A. Zadeh, "From computing with numbers to computing with words–from manipulation of measurements to manipulation of perceptions". IEEE Trans. on Circuits and Systems–I: Fundamental Theory and Applications, vol. 45, no. 1, pp. 105-119, 1999

[7] L. A. Zadeh, "Web Intelligence, World Knowledge and Fuzzy Logic–The Concept of Web IQ (WIQ)", University of California Berkeley, 2004.

[8] VMware Infrastructure 3: Install and Configure. Laboratory Exercise, VMware, EDU-IC-3020-SL-B, 2006

[9] www.vmware.com/vmtn/resources/410, VMware Infrastructure 3 Architecture.

[10] www.vmware.com/pdf/vi3_installation_guide.pdf, Installation and Upgrade Guide.

[11] www.vmware.com/pdf/vi3_server_config.pdf, Server Configuration Guide.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Sense-Based Clustering of Polish Nouns in the Extraction of Semantic Relatedness

Bartosz Broda*, Maciej Piasecki*, Stanisław Szpakowicz†‡
*Institute of Applied Informatics, Wrocław University of Technology, Poland
{bartosz.broda,maciej.piasecki}@pwr.wroc.pl
†School of Information Technology and Engineering, University of Ottawa, Canada
szpak@site.uottawa.ca
‡Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

***Abstract*—The construction of a wordnet from scratch requires intelligent software support. An accurate measure of semantic relatedness can be used to extract groups of semantically close words from a corpus. Such groups help a lexicographer make decisions about synset membership and synset placement in the network. We have adapted to Polish the well-known algorithm of Clustering by Committee, and tested it on the largest Polish corpus available. The evaluation by way of a *plWordNet*-based synonymy test used Polish WordNet, a resource still under development. The results are consistent with a few benchmarks, but not encouraging enough yet to make a wordnet writer's support tool immediately useful.**

## I. Introduction

THE construction of a wordnet for yet another language, especially from scratch, requires a significant effort. There is a high pay-off: wordnets are essential in a fast growing number of applications. One way of reducing the cost is to facilitate wordnet development by automatic tools that suggest missing synsets and relations among them. Two paradigms of extracting semantic relation are cited [1]: based on patterns and based on clustering. The existing methods, however, extract relations between *lexical units* (LUs), while the problem of synset construction has been left open.

*Measures of Semantic Relatedness* (MRSs) based on *distributional semantics* generate a continuum of relatedness values for pairs of LUs. Even a casual look at a list of LUs most related to a given unit $u$ reveals numerous semantic relations: synonymy and antonymy (with similar distributional patterns), hypernymy, meronymy, metonymy, LUs semantically linked to $u$ by some situation type, and so on. Precise annotation of a list of pairs of related LUs with different types of semantic relations is a difficult manual task, and low inter-annotator agreement is likely if the task is performed without context. Any definition of hypernymy, meronymy and in particular synonymy is stated as a textual description that relies on the annotator's language competence and, at best, is supported by a diagnostic substitution test. Semantic and pragmatic constraints make many LUs semantically related to many other LUs. This view of wordnet relations, especially including near-synonymy – the basis for determining synsets – suggests that these relations are just weakly identifiable characteristic subspaces in the continuum of semantic relatedness.

Carefully selected lexico-syntactic patterns can identify pairs of LUs by a hypernymy-type relation, but their coverage (recall) is limited, precision imperfect, and a sharp distinction among remote, indirect hypernyms and more direct close hypernyms, or even near-synonyms, is not possible on a large scale.

MSRs and lexico-syntactic patterns deliver only some clues. The notion of a synset is not exactly constructively defined. To approximate synset structure automatically, several clues may have to be combined. A densely interlinked group of LUs strongly related via a MSR seems to be a natural first approximation.

The main objective of our research is the identification of tightly interlinked groups of words as representing near-synonymy and close hypernymy. Next, on the basis of the extracted groups, we identify LUs represented by lemmas belonging to groups and their sense as described by groups. Finally, we propose to extend a wordnet semi-automatically with the collected synsets and LUs.

Several clustering algorithms have been discussed in literature for the task of grouping LUs. Among them, Clustering by Committee (CBC) [2], [3] has been reported to achieve good accuracy in comparison to *plWordNet*. It is often referred to in the literature as one of the most interesting clustering algorithms, e.g. [4].

CBC relies only on a modestly advanced dependency parser and a MSR based on pointwise mutual information (PMI) extended with a discounting factor [3]. This MSR is a modification of Lin's measure [5] analysed in [6] in application to Polish. Both measures are close to the RWF measure [7] that achieves good accuracy in comparison to Polish WordNet [8].

Our goal was to analyse CBC's applicability to an inflected language for which there is a limited set of language processing tools, and to extract LU groups for the purpose of extending Polish WordNet. We expected to identify several groups of high internal similarity for a polysemous word. Moreover, we wanted to improve CBC's accuracy and to analyse its dependence on several thresholds which are explicitly, but also implicitly, introduced in CBC. We were looking for a more objective and straightforward evaluation of the algorithm results than originaly proposed in [3].

Applications of CBC to languages other than English are rarely reported in the literature. Tomuro et al. [9] mentioned briefly some experiments with Japanese, but gave no results. However, differences between languages, and especially differences in resource availability for different languages, can affect the construction of the similarity function at the heart of CBC. Moreover, CBC crucially depends on several thresholds whose values were established experimentally. It is quite unclear to what extent they can be reused or re-discovered for different languages and language resources.

## II. The CBC Algorithm

The CBC algorithm has been well described by its authors [2], [3]. We will therefore only outline its general organisation, following [3] and emphasising selected key points. We have reformulated some steps in order to name consistently all thresholds present in the algorithm. Otherwise, we keep the original names.

```
I    Find most similar elements
        1) for each word $e$ in the input set $E$, select $k$ most
           similar words considering only $e$'s features
           above the threshold $\theta_{MI}$ of mutual information
II   Find committees
        1) extract a set of unique word clusters by average
           link clustering, one highest-scoring cluster per
           list
        2) sort clusters in descending order and for each
           cluster calculate a vector representation on the
           basis of its elements
        3) going down the list clusters in sorted order,
           extend an initially empty set $C$ of committees
           with clusters similar to any previously added
           committee below the threshold $\theta_1$
        4) for each $e \in E$, if the similarity of $e$ to any
           committee in $C$ is below the threshold $\theta_2$, add
           $e$ to the set of residues $R$
        5) if $R \neq \emptyset$, repeat Phase II with $C$ (possibly $\neq \emptyset$)
           and $E = R$
III  Assign elements to clusters
        • for each $e$ in the initial input set $E$
          1) $S$ = identify $\theta_{T200} = 200$ committees most
             similar to $e$
          2) while $S \neq \emptyset$
             a) find a cluster $c \in S$ most similar to $e$
             b) exit the loop if the similarity of $e$ and $c$
                is below the threshold $\sigma$
             c) if $c$ "is not similar" to any committee in
                $C^1$, assign $e$ to $c$ and remove from $e$ its
                features that overlap with $c$'s features
             d) remove $c$ from $S$
```

CBC has three main phases, marked by Roman numerals above. In the initial Phase I, data expressing semantic similarity of LUs are prepared. Here, CBC shows strong dependency on the quality of the applied MSR – the most important CBC parameter – and the MSR is transformed by taking into consideration only some features (the threshold $\theta_{MI}$) and the $k$ most similar LUs.

In the next two phases, the set of possible senses is first extracted by means of committees; next, LUs are assigned to committees. A *committee* is an LU cluster intended to express some sense by means of a cluster vector representation derived from features describing the LUs included in it. Committees are selected from the initial LU clusters generated by processing the lists of the $k$ most similar LUs, see II.1 and II.2. However, only the groups dissimilar to other selected groups are added to the set of committees, because the committees should ideally describe all senses of the input LUs, see II.3. The set of committees is also iteratively extended in order to cover senses of all input LUs, see the condition in III.4.

Committees only define senses. They are not the final LU groups we are going to extract. The final LU groups – ideally sets of near synonyms – are extracted on the basis of committees in Phase III. Each LU can be assigned to one of several groups on the basis of the similarity to the corresponding committees. It is assumed that each sense of a polysemous LU corresponds to some subset of features which describe the given LU. In step III.2.c, each time a LU $e$ is assigned to some committee $c$ (i.e. the next sense of $e$ has been identified) CBC attempts to identify the features describing the sense $c$ of $e$ and remove them before the extraction of the other senses of $e$ . The idea behind this operation is to remove the sense $c$ from the representation of $e$, in order to make other senses more prominent. However, the implementation of the *overlap* and *remove* operations is straightforward: values of all features in the intersection are simply set to 0 [2]. It would be correct if the association of features and senses were strict, but it is very rarely the case. Mostly, one feature derived from lexico-syntactic dependency corresponds in different amount to several senses. A less radical solution for sense representation removal is proposed in Section V.

## III. CBC Applied to Polish

Our initial intention was to re-implement CBC as published in [2], [3], in order to analyse and compare its performance for Polish. However, we face two problems – there are significant typological differences between the two languages and the availability of language tools differs. For example, unlike English (for which CBC was originally designed), Polish is generally a free word-order language; much syntactic information is encoded by rich inflection. This makes the construction of even a shallow parser for Polish more difficult than for English, e.g. noun modification by another noun is marked by the genitive case, but genitive is also required by negated verbs, and the noun modifier can occur either in a pre-modifying or post-modifying position. On the other hand, there are possibilities of exploring morpho-syntactic relations between word forms (but not in the case of the noun-noun modification). As no verb subcategorisation dictionary is

[1] We interpret this as $c$'s similarity being below an unmentioned threshold $\theta_{ElCom}$.

available for Polish, the identification of verb arguments in text is almost impossible, and semantic description of nouns can be based on relations to verbs only to a small extent.

CBC begins by running a dependency parser on the corpus. No similar tool exists for Polish. In [7], [6] a similar problem was successfully solved by applying several types of lexico-morphosyntactic constraints to identify a subset of structural dependencies mainly on the basis of morphological agreement among words in a sentence and a few positional features like noun-noun sequence of modification. A direct comparison of MSRs based on parsing and on constraints is not yet possible, but the constructed constraint-based MSRs have good accuracy when compared with *plWordNet* [8] by a modified version of *WordNet-Based Synonymy Test* (WBST) [10]. By applying the constructed MSR we got results comparable with the results achieved by humans in the same task [11]. We therefore assumed that the constructed MSR is at least comparable in quality to the one used in [2], [3], and we adopted the constraint-based approach here, applying the same constraints as in [7].

As in [6], [7], the applied constraints are written in the JOSKIPI language and run by the engine of the TaKIPI morphosyntactic tagger [12]. Each noun $n$ is described by the frequency with which occurrences of $n$ in the corpus meet two lexico-morphosyntactic constraints: modification by *a specific adjective* or *an adjectival participle*, and co-ordination with a *a specific noun*.

MRSs and clustering algorithms constructed for Polish can be evaluated on the basis of *plWordNet*, but *plWordNet* is still quite small in comparison to Princeton *plWordNet* (henceforth PWN). It includes mostly general words and lacks many senses for the words described. This complicates the analysis of the evaluation.

All experiments were run on the IPI PAN Corpus [13] (IPIC), the largest annotated corpus of Polish, extended with a corpus of the on-line edition of a Polish daily, 1993-2001) [14] (Rz). The joint corpus (IPIC+Rz) includes about 368 million tokens, around 2.56 times more than the corpus used in [3]. IPIC+Rz, however, is not well balanced: legal and scientific texts are over-represented, so intuitively rare words may have inflated frequencies, but many “popular” words have low frequencies. TaKIPI does not distinguish proper names. Lemmatization makes more errors than it is the case for English.

Several thresholds used in the CBC algorithm (plus a few more in the evaluation) are the major difficulty in its exact re-implementation. Moreover, any method of the CBC optimisation in relation to thresholds was not proposed in [2], [3] and the values of all thresholds were established experimentally in [2]. There also was no discussion of their dependence on the applied tools, corpus and characteristics of the given language. We will discuss the values of most of these thresholds:

- $k$ – the tested value range: $[10, 20]$ [2, p. 53], but the final choice is not given,
- $\theta_{MI}$ – the exact value is not presented, but it is claimed that $\theta_{MI}$ “had no visible impact on cluster quality” [2, p. 53],
- $\theta_1 = 0.35$ [2, p. 55],
- $\theta_2 = 0.25$ [2, p. 55],
- $\theta_{T200} = 200$ [2, p. 58],
- $\sigma$ – different values tested [2, pp. 95-96], while the best score was reported with $\sigma = 0.18$, however, in the chart on pp. 96 of [2] the best result is presented for $\sigma = 0.1$, which we assumed as the default value.

A crucial threshold $\theta_{ElCom}$ – it influences the process of assigning elements to word groups in Phase III – is not overtly named in the algorithm [3], [2]; the values applied to $\theta_{ElCom}$ are unknown. The possibility that $\theta_{ElCom}$ is identical with $\sigma$ is excluded by the order of steps: 2b comes before 2c. For $\theta_{T200}$ no other values were tested but it is reasonably high: it is unlikely ever to have more than 200 senses of a word. Besides the unknown value of $\theta_{ElCom}$, other thresholds seem to depend on the corpus and, especially, on the properties of the MSR.

To extract clusters in Phase II, we applied the CLUTO package [15], which allowed us to analyse the influence of several clustering strategies, namely: *UPGMA*, *i1*, *i2*, *h1*, *slink* and *wclink*, besides the average-link clustering originally applied in CBC. During the first experiment, we used a MSR based on PMI, constructed according to the equations presented in [3]. The results of this experiment appear in Table I.

In the experiments presented in [11], [6], MSR based on Rank Weight Function used for the transformation of feature frequencies generally surpassed several other types of MRS known from the literature, some of them similar to the PMI measure applied in CBC, e.g. see [11], [6]. In the second experiment we replace PMI MSR with RWF MSR.

## IV. Evaluating CBC on Polish

As we wrote in section III, all experiments were run on the IPIC+Rz corpus. We wanted to evaluate the algorithm’s ability to reconstruct *plWordNet* synsets. That would confirm the applicability of the algorithm in the semi-automatic construction of wordnets. We put nouns from *plWordNet* on the input list of nouns ($E$ in the algorithm). Because *plWordNet* is constructed bottom-up, the list consisted of 13298 most frequent nouns in IPIC plus some most general nouns, see [8]. The constraints were parameterised by 41599 adjectives and participles, and 54543 nouns – 96142 features in total.

### A. Evaluating Extracted Word Senses

Evaluation of the extracted word senses proposed in [3], [2] is based on comparing the extracted senses with those defined for the same words in PWN. It is assumed that for a word $w$ a correct sense is described by a word group $c$ such that $w \in c$ if a synset $s$ in PWN such that $w \in s$ is sufficiently similar to $c$. The latter condition is represented by another threshold $\theta$.

The notion central to the evaluation proposed in [3], [2] is similarity between wordnet synsets. The definition of similarity was based on probabilities assigned to synsets and derived from a corpus annotated with synsets. This kind of synset similarity is very difficult to estimate for languages for which there is no such corpus, as is the case of Polish. In order to avoid any kind of unsupervised estimation of synset probabilities, we used a slightly modified version of Leacock's similarity measure[16]:

$$sim(s_1, s_2) = -log(\frac{Path(s_1, s_2)}{\max_{s_a, s_b} Path(s_a, s_b)}), \qquad (1)$$

$Path(a, b)$ is the length of a path between two synsets in *plWordNet*.

Except for synset similarity, we follow [3], [2] strictly in other aspects of word sense evaluation. Synset similarity is used to define the similarity between a word $w$ and a synset $s$. Let $S(w)$ be a set of wordnet synsets including $w$ (its senses). The similarity between $s$ and $w$ is defined as follows:

$$simW(s, w) = max_{t \in S(w)} sim(s, t) \qquad (2)$$

Similarity of a synset $s$ (a sense recorded in a wordnet) and a group of LUs $c$ (extracted sense) is defined as the average similarity of LUs belonging to $c$. However, LU groups extracted by CBC have no strict limits – their members are of different similarity to the corresponding committee (sense pattern). The core of the LU group is defined in [3], [2] via a threshold $\kappa$[2] on the number of LUs belonging to the core. Let also $c_\kappa$ be the core of $c$ – a subset of $\kappa$ most similar members of $c$'s committee. The similarity of $c$ and $s$ is defined as follows:

$$simC(s, c) = \frac{\sum w \in c_\kappa simW(s, u)}{\kappa} \qquad (3)$$

We assume that a group $c$ corresponds to a correct sense of $w$ if

$$max_{s \in S(w)} simC(s, c) \geq \theta \qquad (4)$$

The wordnet sense of LU $w$, corresponding to the sense of $w$ expressed by a LU group $c$, is defined as a synset which maximizes the value in formula 4:

$$arg\, max_{s \in S(w)} simC(s, c) \qquad (5)$$

The question arises why this evaluation procedure is so indirect. Why do we not compare the cores of the LU groups with wordnet synsets? The answer is seemingly simple. Both in Polish and in English, certain matches are hard to obtain. LU groups are indirectly based on the MSR used. They do not have clear limits, and still express some closeness to a sense, but not to a strictly defined sense. On the other hand, wordnet synsets also express a substantial level of subjectivity in their definitions, especially when they are intended to describe *concepts*, which are not directly observable in language data. The proposed indirect evaluation will measure the level of resemblance between the division into senses made by wordnet writers and that extracted via clustering.

As stated previously, the selection of committees is critical, because it affects the remainder of the algorithm. Obviously, the criterion function for agglomerative clustering used in step of Phase II is important in this process. We therefore measure the precision of assigning words to correct sense using different criterion functions. The results appear in Table I. We used default values for thresholds: $\theta_1 = 0.35$, $\theta_2 = 0.25$, $\sigma = 0.1$, $\theta_{MI} = 250$ and $k = 20$. We assumed that default value for $\theta_{ElCom}$ is 0.2. Previous investigation of the properties of RWF [6] revealed that it behaves differently than MSRs based on mutual information. We chose different default values for RWF: $\theta_1 = 0.2$, $\theta_2 = 0.12$. Also, $\theta_{MI}$ does not apply to RWF, so for fair comparison we used another threshold – on the minimal frequency with which a word appears in any relation $min_{tf} = 200$ and on the minimal number of different relation in which the word appeared with $min_{nz} = 10$.

The selection of threshold values was done on the basis of experiments. Automatising this process is a very difficult problem, as the whole process is computationally very expensive – one full iteration takes 5-7 hours on a PC 2.13 GHz and 6 GB RAM, that makes e.g. application of Genetic Algorithms barely possible.

The differences between slink, UPGMA and i2 (see Tab. I) are very small. We have chosen the i2 criterion for further experiments because of its efficiency.

TABLE I
PRECISION FOR DIFFERENT CRITERION FUNCTION OF THE AGGLOMERATIVE CLUSTERING ALGORITHM.

| | PMI | | RWF | |
|---|---|---|---|---|
| | Precision | No. of words | Precision | No. of words |
| UPGMA | 22.59 | 2993 | 38.42 | 682 |
| i1 | 23.45 | 2980 | 35.72 | 744 |
| i2 | 22.37 | 2995 | 38.81 | 742 |
| h1 | — | — | 31.88 | 345 |
| slink | 22.70 | 2982 | 37.59 | 665 |
| wclink | 22.98 | 2981 | 34.14 | 703 |

The comparison – presented in Table I – of the influence on CBC of both MSRs used, PMI and RWF, is a little misleading; in these cases the number of clustered words is very different. This was caused by keeping the same value of the threshold $\sigma = 0.1$ for both versions. It seems that the value of $\sigma$ must be carefully selected for each type of MSR separately.

In Table I we can see that the differences in the algorithm of agglomerative clustering used in generating committees influence the final precision. The best, i2, leads to visibly better committees and word groups.

Because the value of $\sigma$ is so important for the result, we tested its several values with the other parameters fixed (RWF MSR, i2 clustering, $\theta_{ElCom} = 0.2$):

- $\langle \sigma = 0.1$, precision = 38.81, number of words assigned = 742$\rangle$,
- $\langle \sigma = 0.12, P = 40.33, N = 719 \rangle$,
- $\langle \sigma = 0.15, P = 40.99, N = 688 \rangle$,
- $\langle \sigma = 0.18, P = 42.14, N = 655 \rangle$.

With the increasing value od $\sigma$ the precision increases, but the number of words clustered drops significantly. The tendency persists for higher values of both thresholds, e.g. $\langle \theta_{ElCom} = 0.3, \sigma = 0.25, P = 45.4, N = 522 \rangle$. When we set $\sigma$ small and $\theta_{ElCom}$ we get relatively good precision but more words

[2] We changed the original symbol $k$ to $\kappa$ so as not to confuse it with $k$ in the algorithm.

clustered, e.g. $\langle\theta_{ElCom} = 0.3, \sigma = 0.1, P = 38.81, N = 742\rangle$. It means that, contrary to the statement and chart in [2], tuning of both thresholds was important in our case.

In order to illustrate the work of the algorithm, we selected two examples of correct word senses extracted for two polysemous LUs. The word senses are represented by committees described by numeric identifiers. In this way it is emphasised that committee members define only some word sense and are not necessarily near synonyms of the given LU.

LU: **bessa** *economic slump*

id=95 committee:{ niezdolność *inability*, paraliż *paralysis*, rozkład *decomposition*, rozpad *decay*, zablokowanie *blockage*, zapaść *collapse*, zastój *stagnation* }

id=153committee:{ tendencja *tendency*, trend *trend* }

LU: **chirurgia** *surgery*

109 committee:{ biologia *biology*, fizjologia *physiology*, genetyka *genetics*, medycyna *medicine* }

196 committee:{ ambulatorium *outpatient unit*, gabinet *cabinet*, klinika *clinic*, lecznictwo *medical care*, poradnia *clinic*, przychodnia *dispensary* }

Now, the same but with the proposed *heuristic of minimal value activated*, see Section V.

LU: **bessa**

64 committee: {pobyt *stay*, podróż *travel*} – a spurious sense

95 committee: **as above**

153 committee: **as above**

LU: **chirurgia**

109 committee: **as above**

171 committee: {karanie *punishing*, leczenie *treatment*, prewencja *prevention*, profilaktyka *prophylaxis*, rozpoznawanie *diagnosing*, ujawnianie *revealing*, wykrywanie *discovering*, zapobieganie *preventing*, zwalczanie *fight*, ściganie *pursuing, prosecuting*} – a correct additional sense found

196 committee: **as above**

Next, two examples of committees and the generated word groups.

- **committee 57**: {ciemność *darkness*, cisza *silence*, milczenie *silence = not speaking*}
- **LU group**: {cisza, milczenie, ciemność, spokój *quiet*, bezruch *immobility*, samotność *solitude*, pustka *emptiness*, mrok *dimness*, cichość *silence (literary)*, zaduma *reverie*, zapomnienie *forgetting*, nuda *ennui*, tajemnica *secret*, otchłań *abyss*, furkot *whirr*, skupienie *concentration*, cyngiel *trigger*, głusza *wilderness*, jasność *brilliance*}
- **committee 69**: {grota *grotto*, góra *mountain*, jaskinia *cave*, lodowiec *glacier*, masyw *massif*, rafa *reef*, skała *rock*, wzgórze *hill*}
- **LU group**: {góra, skała, wzgórze, jaskinia, masyw, pagórek *hillock*, grota, wzniesienie *elevation*, skałka *small rock*, wydma *dune*, górka *small mountain*, płaskowyż *plateau*, podnóże *foothill*, lodowiec, wyspa *island*, wulkan *volcano*, pieczara *cave*, zbocze *slope*, ławica *shoal*}

Finally, an example of a polysemous committee and the LU group generated on this basis. The group clearly consists of two separate parts: animals and zodiac signs.

- **committee 11**: bestia *beast*, byk *bull*, lew *lion*, tygrys *tiger*
- **LU group**: {lew, byk, tygrys, bestia, wodnik *aquarius*, koziorożec *capricorn*, niedźwiedź *bear*, smok *dragon*, skorpion *scorpio*, nosorożec *rhinoceros*, bliźnię *twin*, lampart *leopard*, bawół *buffalo*}

The last examples clearly show the role of the committee in defining the main semantic axis of the LU group. Two general LUs but semantically different occurring in the same committee makes it ambiguous between at least two senses. Such a committee results in inconsistent LU groups created on its basis. Thus the initial selection of committees is crucial for the quality of the whole algorithm, and the CBC quality depends directly on the MSR applied.

### B. Evaluating by a Synonymy Test

The estimation of synset similarity is not reliable without synset probabilities, at least as the basis of a reimplementation of the evaluation proposed in [3], [2]. We have therefore constructed an additional measure of the accuracy of clustering. We assumed that proper clustering should be able to clear the MSR from accidental or remote associations. That is to say, if two words belong to the same word group, it is a strong evidence of their being near-synonyms or at least being closely related in the hypernymy structure.

In WordNet-Based Synonymy Test (WBST) [10], [11], for each LU $q$ we create a set of four possible answers $A$ in such a way that only one $p \in A$ belongs to the same synset as $q$. The three detractors are selected randomly but do not belong to any synset either of $q$ or $p$. Next, we evaluate the accuracy of choosing $p$ among $A$ on the basis of MSR: we automatically select $max_{a \in A} MSR(q, a)$. In the evaluation of clustering on the basis of WBST we use sequentially two criteria in answering a single WBST question. The results of clustering is the primary criterion, and the MSR is secondary. Here is the algorithm of selecting the answer for a pair $\langle q, A\rangle$:

1) if there is only one $a$ such that $a$ belongs to a LU group of $q$, return $a$
2) if there is a subset $W_A \subseteq A$ whose every element is in one word group with $q$ (not necessarily the same one), for each $a \in W_A$:
   a) calculate the rank position of $rank(a, q)$ in a LU group of $q$ on the basis of similarity to the committee
   b) select subset $W_H R \subseteq W_A$ of elements with the highest rank
   c) if $|W_H R| > 1$, return $max_{a \in W_H R} MSR(a, q)$
3) return $max_{a \in A} MSR(a, q)$

If more possible answers belong to one of the LU groups of $a$, we need to compare them. Each element of a LU group has some similarity to this group's committee, but the similarity values depend on the size of the committee. Committees are represented by centroids calculated from feature vectors of the members. With more members the number of non-zero features increases, and the average values for most features are smaller, so the resulting values of the similarity to the elements of the word group are lower. Instead of the exact similarity values, we arrange all LU group elements in the linear order of their similarity. The resulting ranks are next used in step 2a to compare different possible answers.

If the results of clustering do not give enough evidence to select the answer, we select the answer on the basis of the MSR alone.

We generated 2726 WBST questions from *plWordNet*. The RWF MSR applied alone to solving the test gave 90.97% accuracy (2480 correct and 246 incorrect answers).

TABLE II
ACCURACY IN WBST TEST. SIZE$_{CBC}$ EXPRESSES % OF RESPONSES GIVEN BY CBC.

| | Acc. [%] | Acc., CBC only[%] | CBC q. | Size$_{CBC}$[%] |
|---|---|---|---|---|
| UPGMA | 90.61 | 93.94 | 495 | 18 |
| i1 | 90.68 | 94.30 | 491 | 18 |
| i2 | 90.54 | 94.32 | 493 | 18 |
| h1 | 89.62 | 85.89 | 319 | 12 |
| slink | 90.35 | 93.13 | 466 | 17 |
| wclink | 90.28 | 93.50 | 523 | 19 |

The application of the combined algorithm based on CBC and RWF MSR achieved the accuracy of 90.68% (see Table II). The result of CBC-based algorithm is only slightly worse, but the conclusion is that CBC clustering did not bring any improvement to RWF MSR in its ability to distinguish between a near-synonym and non-related LUs.

In the next experiment we applied RWF MSR and the CBC-based algorithm to solving a (much more difficult) Enhanced WBST (EWBST) proposed in [11]. In EWBST wrong answers are randomly selected from LUs which are *similar* to the proper answer. The similarity is defined on the basis of a wordnet, *plWordNet* in our case. RWF MSR scores 55.52% in EWBST. The result of CBC-based algorithm is significantly lower in EWBST than the result of RWF MSR alone. LU groups generated by CBC include too many loosely related LUs. Assigning a LU to a LU group depends on the similarity to the committee vector and the implicit threshold $\theta_{ElCom}$. Both MSRs generated on our corpus using morphosyntactic constraints can have different levels of values for different lists of the most semantically related LUs. This complicates setting the value of $\theta_{ElCom}$ and generating more consistent word groups.

The results of the evaluation by synonymy test are consistent with the results in IV-A and reveal the source of low precision: loosely related LUs are too often grouped in the same groups. The achieved results of CBC evaluation are in contrast with the better score of RWF MSR alone.

## V. IDENTIFYING SUBSEQUENT SENSES

CBC can assign a LU $w$ to several LU groups, because $w$ can be similar to several committee centroids. It is assumed that the representation of different senses can depend on different features. In order to emphasise the representation of subsequent senses in the vector of $w$, some the features overlapping with the committee centroid $v_c$ are removed from the vector of $w$ in step 2c. We found this technique too radical. We performed a manual inspection of data collected in a co-occurrence matrix. We concluded that it is hard to expect any group of features to encode some sense unambiguously. Moreover, some features have low, accidental values, while some are very high. Finally, vector similarity is influenced by the whole vector, especially when we analyse the absolute values of similarity by comparing it to a threshold, e.g. $\sigma$ in step 2b of CBC.

TABLE III
ACCURACY IN EWBST TEST. SIZE$_{CBC}$ EXPRESSES % OF RESPONSES GIVEN BY CBC.

| | Acc. [%] | Acc., CBC only[%] | CBC q. | Size$_{CBC}$[%] |
|---|---|---|---|---|
| UPGMA | 54.47 | 59.81 | 642 | 24 |
| i1 | 54.55 | 60.82 | 684 | 25 |
| i2 | 54.80 | 62.42 | 660 | 24 |
| h1 | 54.43 | 49.08 | 379 | 14 |
| slink | 54.47 | 60.60 | 637 | 23 |
| wclink | 54.40 | 56.07 | 601 | 22 |

Assuming that a group of features and some part of their ‘strength’ are associated with a sense just recorded, we wanted to look for an estimation of the extent to which feature values should be reduced. The best option seems to be the extraction of some association of features with senses, but for that we need an independent source of knowledge for grouping features, as it was done in [9]. Unfortunately, it is not possible in the case of a language with limited resources like Polish. Instead, we tested two simple heuristics ($w(f_i)$ is the value of the $f_i$ feature, $v_c(f_i)$ – the value of $f_i$ in the committee centroid):

- minimal value – $w(f_i) = w(f_i) - min(w(f_i), v_c(f_i))$,
- the ratio of committee importance – $w(f_i) = w(f_i) - w(f_i)\frac{v_c(f_i)}{\sum v_c(\bullet)}$.

In the minimal value heuristics we make quite a strong assumption that a feature is associated only with one sense on one of the sides: LU and committee. The lower value identifies the right side. The ratio heuristics is based on a weaker assumption: the feature corresponds to the committee description only to some extent.

The application of both heuristics was tested experimentally. We used the settings that resulted in the best precision in Table I, namely RWF MSR, i2 used for initial clustering and the original technique of removing features. The minimal-value heuristics increased the precision from 38.8% to 41% on 695 words clustered. The usage of the ratio heuristic improves the result even further – the precision rises to 42.5% on 701 words clustered. A manual inspection of the results showed that the algorithm tends to produce too many overlapping senses while using the ratio heuristic.

## VI. Conclusions and Further Research

Several explicit and implicit thresholds defined in the algorithm make the re-implementation of CBC difficult. Moreover, most of the thresholds seem to depend on the MSR used and, unfortunately, on the corpus. Any optimisation method would be difficult to apply because of the complexity of the whole CBC process. One full iteration takes 5-7 hours on a PC 2.13 GHz and 6 GB RAM (excluding the initial collection of feature frequencies from the corpus). A method that associates the thresholds with some properties of the corpus or MSR would be necessary. We plan to investigate the ways in which at least a subset of thresholds could be derived from the properties of the used MSR and statistical properties of corpora used for the construction of the MSR.

Our experiments on the application of various clustering algorithms to committee extraction shows the dependency of the whole CBC on this initial step. Moreover, committees often express more than one sense. That results in inconsistent LU groups. Once created, a committee is not verified or amended during the subsequent steps of the algorithm. It would be hard, but some method of committee splitting or verifying could improve the consistency of groups.

The achieved precision is much lower than reported in [2], [3] but quite comparable to that reported for a re-implementation of CBC for English done in [9]. Thus, instead of limited resources for Polish, e.g. lack of a dependency parser and typological differences of Polish in relation to English, we were successful in transferring the method. The achieved accuracy shows the limitations of CBC.

The selection of committees in Phase II is restricted to one committee per a list of related LUs. However, such a list can represent more than one sense in the case of a polysemous LU for which the list was generated.

Infrequent words in the corpus are a serious problem, because they generate high values of MSR with other infrequent words. Committees generated for such words negatively bias the whole CBC algorithm. We achieved better results when we constructed committees only from words that are frequent in the corpus, e.g. $\geq 1000$ occurrences.

The original solution of feature removal when assigning LUs to LU groups seemed to be too simplistic. We considered two simple heuristics of decreasing feature value in extent related to the potential feature correspondence to the sense represented by the committee. Both heuristics resulted in the improvement of the precision of word sense extraction. We will investigate this issue further.

Most senses and LU groups generated by CBC are helpful but of too low accuracy to be a tool willingly used by a fastidious linguist who works on extending a wordnet.

We have identified several key elements in CBC that decide about its accuracy: applied MRS, clustering algorithm used for the identification of committees, identification of feature-sense association together with the algorithm of extraction of subsequent senses from LU description and finally the problem of optimisation of the numerous threshold values. Except the last point, we proposed some solutions to all elements but, while we achieved improvement in all of them, all of them seem to be still open research questions.

## Acknowledgment

Work financed by the Polish Ministry of Education and Science, project No. 3 T11C 018 29.

## References

[1] P. Pantel and M. Pennacchiotti, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations," ACL 2006, Ed. ACL, 2006, pp. 113–120. [Online]. Available: http://www.aclweb.org/anthology/P/P06/P06-1015

[2] P. Pantel, "Clustering by committee," Ph.D. dissertation, Edmonton, Alta., Canada, Canada, 2003, adviser-Dekang Lin.

[3] P. Pantel and D. Lin, "Discovering word senses from text," in *Proc. ACM Conference on Knowledge Discovery and Data Mining (KDD-02)*, Edmonton, Canada, 2002, pp. 613–619.

[4] T. Pedersen, "Unsupervised corpus based methods for wsd," E. Agirre and P. Edmonds, Eds. Springer, 2006, pp. 133–166.

[5] H. Li, "A probabilistic approach to lexical semantic knowledge acquisition and structural disambiguation," Ph.D. dissertation, Graduate School of Science of the University of Tokyo, 1998.

[6] B. Broda, M. Derwojedowa, M. Piasecki, and S. Szpakowicz, "Corpus-based semantic relatedness for the construction of polish wordnet," in *Proc. 6th Language Resources and Evaluation Conference (LREC'08)*, 2008, to appear.

[7] M. Piasecki, S. Szpakowicz, and B. Broda, "Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns," in *Proc. Text, Speech and Dialog 2007 Conference*, ser. LNAI, vol. 4629. Springer, 2007.

[8] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda, "Words, concepts and relations in the construction of Polish WordNet," in *Proc. Global WordNet Conference, Seged, Hungary January 22-25 2008*, A. Tanács, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen, Eds. University of Szeged, 2008, pp. 162–177.

[9] N. Tomuro, S. L. Lytinen, K. Kanzaki, and H. Isahara, "Clustering using feature domain similarity to discover word senses for adjectives," in *Proc. 1st IEEE International Conference on Semantic Computing (ICSC-2007)*. IEEE, 2007, pp. 370–377.

[10] D. Freitag, M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, and Z. Wang, "New experiments in distributional representations of synonymy." in *Proc. Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 25–32.

[11] M. Piasecki, S. Szpakowicz, and B. Broda, "Extended similarity test for the evaluation of semantic similarity functions," in *Proc. 3rd Language and Technology Conference, October 5-7, 2007, Poznań, Poland*, Z. Vetulani, Ed. Poznań: Wydawnictwo Poznańskie Sp. z o.o., 2007, pp. 104–108.

[12] M.Piasecki, "Handmade and automatic rules for Polish tagger," ser. Lecture Notes in Artificial Intelligence, P. Sojka, I. Kopeček, and K. Pala, Eds. Springer, 2006.

[13] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science PAS, 2004.

[14] D. Weiss, "Korpus Rzeczpospolitej," [on-line] http://www.cs.put.poznan.pl/dweiss/rzeczpospolita, 2008, corpus of text from the online edtion of Rzeczypospolita.

[15] G. Karypis, "CLUTO a clustering toolkit," Department of Computer Science, University of Minnesota, Technical Report 02-017, 2002. [Online]. Available: http://www.cs.umn.edu/~cluto

[16] E. Agirre and P. Edmonds, Eds., *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# An immune approach to classifying the high-dimensional datasets

Andrzej Chmielewski
Faculty of Computer Science
Białystok Technical University
Wiejska 45a, 15-351 Białystok, Poland
Email: achmielewski@wi.pb.edu.pl

Sławomir T. Wierzchoń
Institute of Computer Science
Polish Academy of Sciences
and
Faculty of Mathematics, Physics and Informatics
Gdańsk University, Poland
Email: stw@ipipan.waw.pl

***Abstract*—This paper presents an immune-based approach to problem of binary classification and novelty detection in high-dimensional datasets. It is inspired by the negative selection mechanism, which discriminates between self and nonself elements using only partial information. Our approach incorporates two types of detectors: binary and real-valued. Relatively short binary receptors are used for primary detection, while the real valued detectors are used to resolve eventual doubts. Such a hybrid solution is much more economical in comparison with "pure" approaches. The binary detectors are more faster than real-valued ones, what allows minimize computationally and timely complex operations on real values. Additionally, regardless of type of encoding, the process of sample's censoring is conducted with relatively small part of its attributes.**

## I. Introduction

NATURAL Immune System (NIS) prevents organism against intruders called *pathogens*. It consists of a number of cells, tissues, and organs that work together to protect the body. The main agents responsible for the adaptive and learning capabilities of the NIS are white blood cells called lymphocytes and produced by the thymus, spleen and bone marrow.

There are two kinds of lymphocytes: B- and T-lymphocytes called also B- and T-cells for brevity. Lymphocytes start out in the bone marrow and either stay there and mature into B-cells (this process is called *affinity maturation*), or they leave for the thymus gland, where they mature into T-cells. Both the types of lymphocytes have separate jobs to do: T-lymphocytes are like the body's military intelligence system, seeking out their targets and sending defences to lock onto them. B-cells are like the soldiers, destroying the invaders that the intelligence system has identified. It is obvious, that lymphocytes have to tolerate own cells of the organism; otherwise the organism could be self-destroyed.

To better imagine, how difficult and complex is the task of proper classification of different cells and microorganisms, we can mention that the total number of various types of pathogens is far greater then $10^{16}$, whereas there are "only" about $10^6$ own cells types. Despite of such huge diversity of infectious organisms, we require, that efficiency of immune systems has to be at the very high level as every intrusion can case disease or even can lead to die. Moreover, what is a characteristic feature of NIS, T-lymphocytes are "learned" only on the own cells, without any examples of pathogens. It means, that even never seen intruders can be properly recognized as they are different from own cells. This process called *negative selection* is only a small part of complex learning process performed by NIS. A reader interested in detailed description of immune system is referred to e.g. [12], [18].

Mentioned features of NIS are very desirable in many domains such as: intrusion detection systems (IDS), computer viruses detection, novelty detection and other binary classification purposes. Especially, a defending of computer networks against various types of attackers seems to be a natural application domain for immune based algorithms as there is a relatively easy to find out similarities between them. Both systems should effectively, with very high efficiency, recognize undesirable objects (binary classification) which constantly harass them. The huge diversity of intruders, additionally described by significant (from the computational complexity point of view) number of various types of attributes (labels, real and integer values, etc.), makes this problems relatively difficult to solve.

The approach presented in this paper was designed mainly for high-dimensional datasets. It is an effect of many experiments performed with the use of various types of detectors [9], [10], [4] and various affinity measures [3]. At the cost of lengthen learning stage (what is not crucial from e.g. IDS point of view), there is possible to increase the efficiency and significantly speed up the classification process.

This paper is organized as follows. Section II presents current state in this domain. Section III presents the proposed approach to classifying the high-dimensional datasets. Conducted experiments and its results are described in Section IV. Section V concludes the results and discusses a possibility of application of presented approach for other domains.

## II. Background

### A. Negative Selection

The *negative selection* algorithm, proposed by Forrest et. al, [6], is inspired by the process of thymocytes, i.e. young T-lymphocytes, maturation. It is designed to discriminate between own cells (called *self*) and others (called *nonself*).

978-83-60810-14-9/08/$25.00 © 2008 IEEE

To be more formal, denote $U$ the problem space, or Universe of discourse and let $S \subset U$ be a subset of elements (e.g. measurements represented as binary strings or real-values) representing typical behavior of a system under considerations (*self*). Then the set of elements characterizing anomalous behaviour, $N$ can be viewed as the set-theoretical complement of $S$ (*nonself*):

$$N = U \backslash S \tag{1}$$

The negative selection algorithm relies upon generation of so-called *detectors*, or *receptors*, being a counterpart of T-lymphocytes, in such a way, that a freshly generated detector $d$ is added to the set $D$ of valid detectors only if it does not recognize any self element. In the simplest case the detectors are generated randomly, but smart techniques are requested in general [6].

To mimic the process of self/nonself recognition we must designate an affinity measure, $match(d, u)$, specifying the degree with which a detector $d$ bonds a given element $u$, see e.g. [18] for details. Usually, $match(d, u)$ is modeled by a distance metric or a similarity measure [7]. Majority of detection rules induce so-called *holes* ($H$), i.e. regions of $N$ which are not covered by any detector and therefore samples from this region will be classified as self.

In real-life applications, it is not possible to gather all elements of $S$, because typically only its subset is observed. Therefore, it is assumed, that $S$ is composed of $S_{seen}$ and $S_{unseen}$

$$S = S_{seen} \cup S_{unseen}, \tag{2}$$

and only $S_{seen}$ is taken into the training phase (i.e. the phase of detector's generation). As a result, the detectors from the set $D$ can recognize not only nonselfs from $N' \subset N$, but also some elements from $S_{unseen}$ and it can result in wrong classification of not seen self samples.

In Fig. 1, there are presented possible combination of $S$, $N$ and $H$ regions. In Fig. 1(a) there is no holes as the self samples fully covers a space $S$ and a detectors fully covers $N$ region. This is only theoretical situation, not met in real applications. The desirable situation is in Fig. 1(b); detectors not covers $S_{unseen}$ region. In this case, holes plays an important role as they are *"necessary to generalize beyond the training set"* (Stibor [15]). Problems of overfitting and underfitting are showed in Fig. 1(c) and Fig. 1(d), respectively. In both cases, a detectors covers either too much or too low of space, what leads to decreasing the accuracy of negative selection algorithms.

### B. Optimal repertoire of detectors

In negative selection, the time complexity of classification as well as space complexity is relevant to the number of generated detectors. Every censored sample, in pessimistic case, have to be matched with all detectors from $D$. Thus, to decrease the complexity of classification process, it is desired, to generate minimal number of detectors covering a maximal part of $N$. A perfect set of detectors should cover themselves in minimal way and then each of them can be able to recognize different subsets of $N$ (optimal repertoire of detectors).

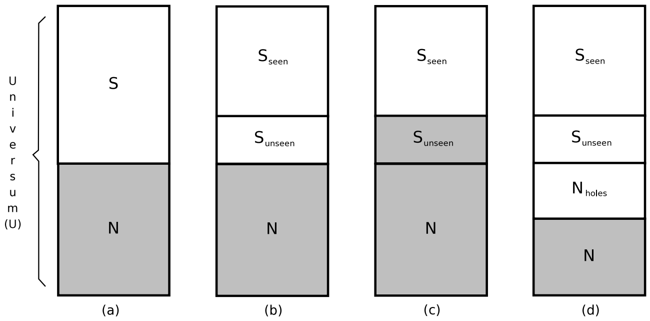


Fig. 1. Examples of possible combination of Self($S$), Nonself($N$) subsets, holes and regions covered by detectors (greyed regions).

This problem is especially important in on-line systems where huge dataset have to be classified without significant delays (i.e. IDS, virus detection, etc.).

### C. Representing samples

To model the interactions (matching) between self and nonself samples we need three elements:

- proper samples encoding,
- appropriate affinity function (matching rule), and
- appropriate algorithm enabling generation of the receptors.

The choice of appropriate representation for samples gathered in the datasets seems to be a key issue. Typically, most of the samples are encoded as vectors of real-valued numbers. Unfortunately, this type of representation can involve the use of many time consuming operations (like multiplication and division), e.g. when classification process have to compute a distance between two samples to measure the degree of similarity between them. Therefore, in the case of huge dataset containing tens of attributes, the choice of this type of representation seems to be far from optimal, especially when the time of classification is crucial.

Since there, two types of samples representation were regarded in negative selection algorithms: binary and real-valued.

*1) Binary representation:* Primary, this type of representation was applied by Forrest et al. [5] to capture anomaly sequences of system calls in UNIX systems and next to model the system for monitoring TCP SYN packets to detect network traffic anomalies (called LISYS) [8].

In case of binary encoding, we identify the Universe with $l$-dimensional Hamming space, $\mathbb{H}^l$ consisting of the binary strings of fixed length $l$:

$$\mathbb{H}^l = \{\underbrace{000...000}_{l}, \underbrace{000...001}_{l}, \ldots, \underbrace{111...111}_{l}\}$$

Hence the size of this space is $2^l$. The most popular matching rules used in this case are:

(a) $r$-contiguous bit rule [11], or
(b) $r$-chunks [2].

Both the rules say that a detector bonds a sample only when both the strings contain the same substring of length $r$. To detect a sample in case (a), a window (part of detector) of length $r$ ($0 \leq r \leq l$) is slided through censored samples of length $l$. In case (b) the detector specifies substring and its position within a string. In both the cases nonself samples are recognized based only on partial information.

Below an example of matching ($r$-contiguous rule) a sample by a detector for affinity threshold $r = 3$ is given:

$$\begin{array}{ll} \overbrace{1\;0\;\mathbf{0}\;\mathbf{0}\;\mathbf{1}\;1\;1\;0}^{l} & \text{sample} \\ 0\;1\;\underbrace{\underline{\mathbf{0}\;\mathbf{0}\;\mathbf{1}}}_{r}\;0\;0\;1 & \text{detector} \end{array}$$

It is obvious, from optimal repertoire point of view (see Section II-B), that shortest detectors are more desirable as they are able to detect more samples. However, Stibor [16] showed the coherence between $r$ and $l$ values for various $|S|$, in the context of the probability of generating detectors. He distinguished three phases:

- Phase 1 (for lowest $r$) – the probability is very near to 0,
- Phase 2 (for middle $r$) – the probability rapidly increase from 0 to 1 (so called *Phase Transition Region*),
- Phase 3 (for highest $r$) – the probability is very near to 1.

Hence, we should be interested in generating receptors with medium length $r$ (belonging to second region) and eventually with larger values of $r$ if coverage of $N$ is not sufficient. It is worth to emphasize, that detectors can not be too long, due to exponential increase in the duration of learning process, which should be finished in reasonable time.

Hofmeyr in [8] developed immune-based IDS, called LISYS, using binary representation and $r$-contiguous rule as a matching function. Detectors, were randomly generated and each network TCP SYN packet was encoded as a binary string of length 49 (32-bits external IP address, 8-bits local address, 8-bits type of service and 1 bit for indicating the server machine). However, he used only 4 attributes which, in the most cases, are not sufficient to detect intruders. For example, Snort [13], one of the most popular open source IDSs, involve the use several (or even tens) parameters at least, significantly lengthen the dimension of space $l$.

*2) Real-valued representation:* To overcome scaling problems inherent in Hamming space, Ji and Dasgupta [10] proposed real-valued negative selection algorithm, termed as *V-Detector*.

It operates on (normalized) vectors of real-valued attributes; each vector can be viewed as a point in the $d$-dimensional unit hypercube, $U = [0, 1]^d$. Each self sample, $s_i \in S$, is represented as a hypersphere centered at $c_i \in U$ and constant radius $r_s$, i.e. $s_i = (c_i, r_s)$, $i = 1, \ldots, l$, where $l$ is the number of self samples. Every point $u \in U$ which lies within any self hypersphere $s_i$ is considered as a self element. Also, detectors $d_j$ are represented as hyperspheres: $d_j = (c_j, r_j)$, $j = 1, \ldots, p$ where $p$ is the number of detectors. In contrast to self elements, the radius $r_j$ is not fixed but is computed as the Euclidean distance from a randomly chosen center $c_j$ to the nearest self element (this distance must be greater than $r_s$, otherwise detector is not created). Formally, we define $r_j$ as

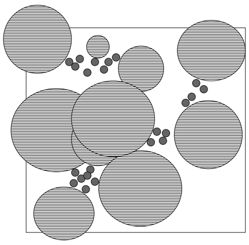


Fig. 2. Example of performance V-Detector algorithm for 2-dimensional problem. Grey circles denotes self samples, dashed circles denotes V-detectors, dashed area denotes detector which recognize all samples laying outside the space spanned over all self samples and white areas denotes holes.

$$r_j = \min_{1 \leq i \leq l} dist(c_j, c_i) - r_s \tag{3}$$

The algorithm terminates if predefined number $p_{max}$ of detectors is generated or the space $U \backslash S$ is sufficiently well covered by these detectors; the degree of coverage is measured by the parameter $co$ – see [10] for the algorithm and its parameters description.

In its original version, the V-Detector algorithm employs Euclidean distance to measure proximity between each two samples. Therefore, self samples and the detectors are hyperspheres (see Figure 2). Formally, Euclidean distance is a special case of Minkowski norm ($L_m$, where $m \geq 1$), which is defined as:

$$L_m(\mathrm{x}, \mathrm{y}) = \left( \sum_{i=1}^{d} |x_i - y_i|^m \right)^{\frac{1}{m}}, \tag{4}$$

where $\mathrm{x} = (x_1, x_2, \ldots, x_d)$ and $\mathrm{y} = (y_1, y_2, \ldots, y_d)$ are points in $\Re^d$. Particularly, $L_2$-norm is Euclidean distance, $L_1$-norm is Manhattan distance, and $L_\infty$ is Tchebyshev distance.

However, Aggarwal et. al [1] observed that $L_m$-norm loose its meaningfulness of proximity distance when the dimension $d$ and the values of $m$ increase. Thus, for example, Euclidean distance is the best (among $L_m$-norms) similarity metric when $d$ is about 5. For higher $d$, metrics with lowers $m$ (i.e. Manhattan distance) should be used.

Based on this observation, Aggarwal introduced *fractional distance metric* with $0 < m < 1$, arguing that such a choice is more appropriate for high-dimensional spaces. Experiments, reported in [3], partially confirmed efficiency of this proposition. For $1 < m < 0.5$, more samples were detected, in comparing to $L_2$ and $L_1$ norms. However, for $m < 0.5$ efficiency rapidly decreased and for $m = 0.2$, none samples were detected. Moreover, the same experiments showed also a trade-off between efficiency, time complexity and $m$. For fractional norms, the algorithm runs slower for lowers $m$;

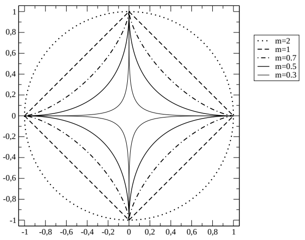


Fig. 3. Unit spheres for selected $L_m$ norms in 2D.

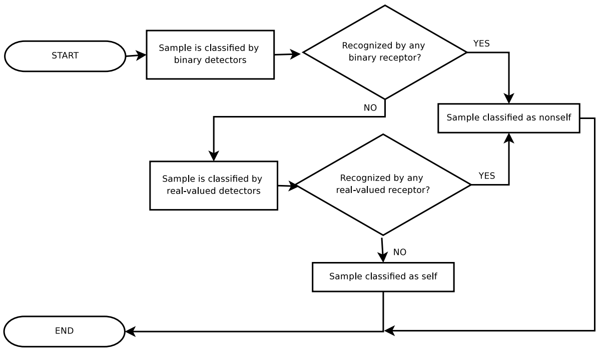

START
Sample is classified by binary detectors
Recognized by any binary receptor?
YES
NO
Sample classified as nonself
Sample is classified by real-valued detectors
Recognized by any real-valued receptor?
YES
NO
Sample classified as self
END

Fig. 4. Scheme of the classification process for dual representation receptors.

duration of learning processes for $L_{0.5}$ were even 2-3 times longer than for $L_2$.

One of the additional consequence of applying fractional metrics for V-Detector algorithm is the change of the shape of detectors. Figure 3 presents the unit spheres for selected fractional $L_m$-norms in 2D.

## III. Proposed approach

In [4] we proposed to use the dual representation of samples. The main motivation was to reduce duration of classification process as well as to improve the detection rate, especially in the case of high-dimensional datasets. When we attain this goal, then immune algorithms will be even more attractive alternative to traditional, i.e. *not immune* approaches in many domains i.e. computer security, spam detection, etc.

For example, IDSs which can represent on-line classification systems, should be able to analyze all network connections, regardless of intensity network traffic. The large number of various parameters which should be taken into consideration and still increasing the number of signatures of potential attackers, requires the use of computers with higher and higher computational power. Thus, signatures of normal (legal) behavior, offered by *negative selection*, seems to be reasonable solution, especially, when more effective models and algorithms will be developed.

The scheme of dual receptors is presented in Figure 4. It involves generating both types of detectors. Binary detectors, plays only the role of preliminary detection and those samples which are not recognized by them are censored by V-Detector algorithm. Thus, we do not expect that binary receptors covers the space $N$ in very high degree, as it can takes too much time. More important aspect is its length. They should be relatively short (with high generalization degree) to detect in the possible shortest time as much as possible nonself samples and then the profits of this approach will increasing. The optimal length of binary detectors $r$ can be easily tuned. Usually, it is placed at the end of *phase transition region* (see Section II-C1). Then, both probability of detecting detectors as well as the corresponding to it coverage of $N$ space are sufficiently high.

To construct binary receptors, after normalization to unitary hypercube, self samples should be turned into binary representation. For every attribute $x$ ($x \in [0, 1]$) the quantization function $Q$ can be expressed as:

$$Q(x) = \lfloor M * x \rfloor, \tag{5}$$

where $\lfloor\cdot\rfloor$ is floor function and $M$ is the number of ranges (clusters). In our case, the optimal number of clusters is $M = 2^{bpa}$, for $bpa = \{1, 2, \ldots\}$ ($bpa$ - bits per attribute).

In [4] and [3] we conducted experiments on datasets containing about 30-40 attributes. For such dimensionality, Euclidean and Manhattan distance metrics provide too low detection rates, what was confirmed by performed experiments. As could be expected, fractional distances produced quite good results (about 70-80%), but we should keep in mind that, in comparison to $L_2$ and $L_1$ norms, they are much more computationally complex.

Unfortunately, there are many datasets which consist of 50 and more attributes (even up to 250), i.e. spam detection, handwritten recognition and even time series. In this case, probably none of known metrics is able to measure the similarity properly when the whole set of attributes is used. Thus, in this paper, we propose to incorporate the sliding window for both binary as well as real-valued detectors.

Let $w$ ($w \leq d$) be the window's length for real-valued detectors. Then, to censor a sample by only one detector, there is a need to calculate $d - w + 1$ distances (instead of 1 in original case), in the pessimistic case. Hence, for example, if $d = 100$, $w = 20$ and $|D| = 1000$, even about $81000$ such operation can be needed to determine, if a sample belong to set $S$ or $N$. Similarly, process of detector's maturation, which depends on $d$, $w$ and $S$, will be also much more durable. Thus, especially for very high-dimensional datasets, binary receptors plays very important role in proposed model.

## IV. Experiments and Results

This section presents only a small part of conducted experiments with most valuable results. Several datasets (presented in I) from Machine Learning Repository and Keogh's were used. For all dataset relatively shorts binary receptors were generated with lengths $l = \{5, 7, 10, 12, 15, 17\}$

For all datasets, relatively shorts binary receptors were generated with lengths $r = \{5, 7, 10, 12, 15, 17\}$ and $bpa = \{1, 2, 3\}$ to examine its usefulness in our model. Selected results are presented in Table II.

One can see, that binary detectors allow to at least 30-40% detection of nonself samples. Moreover, for properly tuned values $r$ and $bpa$, its efficiency increases even up to 80-100%

TABLE I
BRIEF DESCRIPTION OF USED DATASETS.

| Dataset | Number of attr. | Samples count | Self count |
|---|---|---|---|
| Spambase | 54 | 4601 | 2788 |
| Madelon | 250 | 2000 | 1000 |
| Leaf | 160 | 221 | 33 |
| CBF_B | 128 | 5000 | 1723 |
| KDD_ICMP | 41 | 11910 | 4427 |

TABLE II
PERCENT OF NONSELF SAMPLES RECOGNIZED BY BINARY RECEPTORS FOR VARIOUS $r$ AND $bpa$ VALUES.

| Dataset | *bpa* | Rcp. len. | Rcp. count | Det. [%] |
|---|---|---|---|---|
| Spambase | 1 | 10 | 267 | 6.81 |
| | | 12 | 2353 | 10.58 |
| | | 15 | 29015 | 24.10 |
| | | 17 | 126029 | 33.57 |
| Madelon | 1 | 10 | 50 | 2.50 |
| | | 12 | 834 | 16.10 |
| | | 15 | 18530 | 68.60 |
| | | 17 | 100012 | 95.50 |
| Leaf | 1 | 5 | 10 | 21.27 |
| | | 7 | 94 | 55.85 |
| | | 10 | 974 | 75.00 |
| | | 15 | 32620 | 79.25 |
| | 2 | 7 | 75 | 90.95 |
| | | 10 | 875 | 96.80 |
| | | 15 | 32355 | 98.93 |
| | 3 | 12 | 3540 | 100.00 |
| CBF_B | 1 | 12 | 259 | 34.26 |
| | | 15 | 18804 | 54.28 |
| | | 17 | 109255 | 58.22 |
| | 2 | 12 | 2754 | 72.26 |
| | | 15 | 29078 | 81.84 |
| | | 17 | 124673 | 88.98 |
| KDD_ICMP | 1 | 12 | 259 | 34.26 |
| | | 15 | 32536 | 42.02 |

for almost all tested datasets. Thus, one can expected, that such results, should significantly lighten the real-valued detectors.

Next, there was investigated the possibility of applying real-valued receptors for testing datasets. V-Detector algorithm was executed with the following parameters (like in all the experiments described in this section):

- window's length $w = \{5, 10, 15, 20, 25, 30\}$,
- distance norms: $L_2$, $L_1$, $L_{0.7}$, $L_{0.5}$,
- detector's radius $d_r = 0.001$,
- estimated coverage $co = 99.99\%$,
- all self samples were used in learning stage to generate receptors. In consqequence, false positives (number of incorrectly classified self samples) is equal 0 in all cases.

Figures 5 and 6 shows the detection rates for Spambase and Madelon datasets, respectively. It is worth to emphasize, that in our case, detection rates denotes only the number of correctly recognized nonself samples, without correctly classified all selfs. Thus, is difficult to compare obtained results with other approaches, which incorporate all types of samples in at the learning stage.

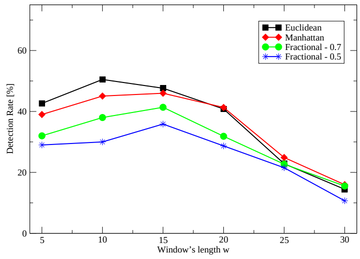


Fig. 5. Spambase. Detection rates for V-Detector algorithm.

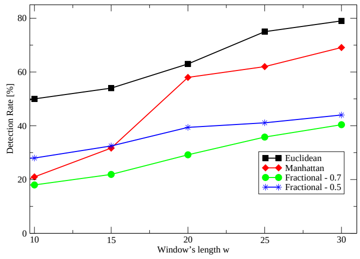


Fig. 6. Madelon. Detection rates for V-Detector algorithm.

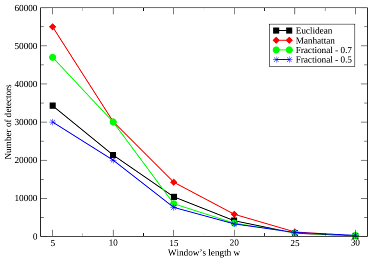


Fig. 7. Spambase. Number of detectors generated by V-Detector algorithm.

Moreover, one can see, for Spambase dataset Euclidean distance gives an optimal results when $w = 10$; for other, considered in this paper similarity metrics, higher efficiency is observed for $w = 15$. It is consistent with presented theoretical reflections [1] and experimental results [3]. On the other hand, for Madelon dataset, detection increase with $w$ (up to 80%).

The duration of classification process, in the case of V-Detector algorithm is proportional to the cardinality of receptor's set. Also, in this case, obtained results are different for presented datasets. For Spambase, number of detectors decreases when $w$ increase. However, for Madelon dataset, we can observe, that for $w = 15$ number of receptors is lowest. In contrast to Spambase, it does not decreases for high $w$ values, what results in very high detection rates.

Finally, the best combinations of parameters for both types of detectors were chosen to achieve the most optimal solutions for our model, considered in this paper. Table III presents the most valuable results. For most testing datasets results are satisfactory. Only for Spambase, detection of *nonself* samples is only slightly higher than 60%.

## V. CONCLUSION

The model presented in this paper base on the negative selection mechanism, developed within domain of Artificial

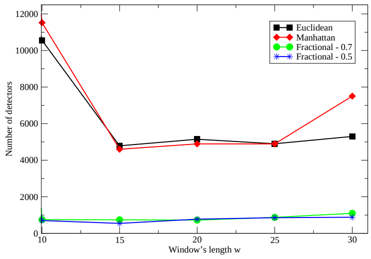


Fig. 8. Madelon. Number of detectors generated by V-Detector algorithm.

TABLE III
THE MOST VALUABLE DETECTION RATES FOR SELECTED DATASETS.

| Dataset | $bpa$ | $r$ | $m$ | $w$ | Det.[%] |
|---|---|---|---|---|---|
| Spambase | 1 | 12 | 1 | 15 | 45.91 |
| | | 15 | 1 | 15 | 49.10 |
| | | 17 | 1 | 15 | 61.10 |
| | 2 | 12 | 1 | 15 | 45.15 |
| | | 17 | 1 | 15 | 47.20 |
| | | 12 | 2 | 10 | 50.28 |
| | | 15 | 2 | 10 | 50.71 |
| Madelon | 1 | 12 | 2 | 20 | 71.45 |
| | | 15 | 2 | 20 | 88.10 |
| | | 17 | 2 | 20 | 98.50 |
| Leaf | 1 | 10 | 2 | 15 | 100.00 |
| | 2 | 10 | 2 | 15 | 100.00 |
| CBF_B | 2 | 12 | 1 | 15 | 100.00 |
| | | 12 | 2 | 15 | 100.00 |
| KDD_ICMP | 1 | 10 | 2 | 10 | 86.61 |
| | | 10 | 0.7 | 10 | 87.99 |

Immune Systems. It is designed for high-dimensional datasets, which are very difficult to classify, due to the lack of appropriate similarity metrics. Sliding window method applied for both types of detectors can be viewed as the one of the possibility to overcome the scaling problem. However, although the detection rates where quite satisfactory, duration of learning as well as classifying processes leave a lot to be desired. Fortunately, significant (and in some cases almost all) part of nonself samples can be relatively quickly detected by binary receptors, which are the power of this model.

Generally, applying only one type of receptors is not able to guarantee the sufficient coverage of nonself space $N$ in reasonable time, especially, when dataset contains a thousands and more samples. Moreover, even slight increase in one of the following set of parameters: $d$, $w$ and $|S|$, significantly influence on the performance of applied algorithms.

Obtained results are very preliminary and should be confirmed with other datasets in the future, regardless of its applications. Moreover, in our opinion, presented model can be, almost directly, applied to detect the various types of anomalies in time series. And it should be a direction of our future investigations.

## ACKNOWLEDGMENT

This work was partly supported by Technical University of Bialystok grant W/WI/1/08.

Experiments were performed on a compute cluster at Faculty of Computer Science, Technical University of Bialystok.

## REFERENCES

[1] Aggarwal C. C., Hinneburgand A., Keim D. A., *On the Surprising Behavior of Distance Metrics in High Dimensional Space*, Lecture Notes in Computer Science, Vol. 1973, 2001, pp. 420–434.

[2] Balthrop J., Esponda F., Forrest S., Glickman M., *Coverage and generalization in an artificial immune system*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, 9-13 July 2002, pp. 3–10.

[3] Chmielewski A., Wierzchoń S. T., *On the distance norms for multidimensional dataset in the case of real-valued negative selection application*, Zeszyty Naukowe Politechniki Białostockiej, No. 2, 2007, pp. 39–50.

[4] Chmielewski A., Wierzchoń S. T., *Dual representation of samples for negative selection issues*, Computer Assisted Mechanics and Engineering Sciences, Vol. 14, No. 4, 2007, pp. 579–590.

[5] Forrest S., Hofmeyr S. A., Somayaji A., Longstaff T. A., *A Sense of Self for Unix Processes*, Proceedinges of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, 1996, pp. 120–128.

[6] Forrest S., Perelson A., Allen L., Cherukuri R., *Self-nonself discrimination in a computer*, in Proceedings of the IEEE Symposium on Research in Security and Privacy, Los Alamitos, 1994, pp. 202–212.

[7] Harmer P. K., Wiliams P. D., Gunsch G. H., Lamont G. B., *Artificial immune system architecture for computer security applications*, IEEE Transactions on Evolutionary Computation, Vol. 6, 2002, pp. 252–280.

[8] Hofmeyr S. A., *An Immunological Model of Distributed Detection and its Application to Computer Security*, Ph.D. thesis, Department of Computer Sciences, University of New Mexico, 1999.

[9] Hofmeyr S. A., Forrest S., *Architecture for an artificial immune systems*, Evolutionary Computation, Vol. 8(4), 2000, pp. 443–473.

[10] Ji Z., Dasgupta D., *Real-valued negative selection algorithm with variable-sized detectors*, Genetic and Evolutionary Computation GECCO-2004, Part I, LNCS Vol. 3102, Seattle, WA, USA, Springer-Verlag, 2004, pp. 287–298.

[11] Percus J. K., Percus O. E., Perelson A. S., *Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination*, Proceedings of National Academy of Sciences USA (90), 1993, pp. 1691–1695.

[12] Perelson A., Weisbuch D., *Immunology for physicists*, Reviews of Modern Physics, Vol. 69, 1977, pp. 1219–1265.

[13] Snort, Intrusion Detection System, http://www.snort.org.

[14] Stibor T., Mohr P., Timmis J., Eckert C., *Is Negative Selection Appropriate for Anomaly Detection?*, in Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO 2005), Washington, D.C., June 25-29, 2005, pp. 321–328.

[15] Stibor T., Timmis J., Eckert C., *The Link between r-contiguous Detectors and k-CNF Satisfiability*, Part of the World Congress On Computational Intelligence, Canada, 2006, pp. 491–498.

[16] Stibor T., *Phase Transition and the Computational Complexity of Generating r-contiguous Detectors*, Proceedings of 6th International Conference on Artificial Immune Systems, LNCS (4628), 2007, pp. 142–155.

[17] Weber A., Schek H. J., Blott S., *A quanttitative analysis and performance study for similarity-search methods in high-dimensional spaces*, 24th International Conference Very Large Data Bases, 1998, pp. 194–205.

[18] Wierzchoń S. T., *Sztuczne systemy immunologiczne*, Teoria i zastosowania. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Application of Clustering and Association Methods in Data Cleaning

Lukasz Ciszak
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665
Warszawa, Poland
Email: L.Ciszak@ii.pw.edu.pl

*Abstract*—**Data cleaning is a process of maintaining data quality in information systems. Current data cleaning solutions require reference data to identify incorrect or duplicate entries. This article proposes usage of data mining in the area of data cleaning as effective in discovering reference data and validation rules from the data itself. Two algorithms designed by the author for data attribute correction have been presented. Both algorithms utilize data mining methods. Experimental results show that both algorithms can effectively clean text attributes without external reference data.**

## I. Introduction

NOWADAYS , information is the most important asset of the majority of companies. Well-maintained information systems allow the company to make successful business decisions. On the other hand, if the information is incomplete or contains errors, it may lead the company to financial loss due to incorrect strategic or tactical decisions.

The purpose of this article is to present current situation in the area of data cleaning and discuss possibilities of application data mining methods in it. The structure of this article is following: first chapter focuses on the area of data quality and data cleaning and presents a categorization of data quality issues. In its last section related work and research in this area is briefly presented. The second chapter of this article contains a discussion of possible applications of data mining methods in the area of data cleaning. The core of this chapter is the presentation of two heuristic data cleaning algorithms designed by the author that utilize a data mining approach. This chapter also contains the results of the experiments performed using the algorithms and a discussion of possible improvements.

### *A.* Data Quality and Data Cleaning

As information is defined as data and method for its interpretation, it is only as good as the underlying data. Therefore, it is essential to maintain data quality. High quality data means that it is “fit for use”[11] and good enough to satisfy a particular business application. The following data quality measures allow one to quantify the degree to which the data is of high quality, namely:

- completeness: all the required attributes for the data record are provided,
- validity: all the record attributes have values from the predefined domain,
- consistency: the record attributes do not contradict one another; e.g. the ZIP code attribute should be within the range of ZIP codes for a given city,
- timeliness: the record should describe the most up-to-date state of the real-world object it refers to. Moreover, the information about an object should be updated as soon as the state of the real world object changes,
- accuracy: the record accurately describes the real world object it refers to; all the important features of the object should be precisely and correctly described with the attributes of the data record,
- relevancy: the database should contain only the information about the object that are necessary for the purpose they were gathered for,
- accessibility and interpretability: the metadata describing the sources of the data in the database and transformations definitions it has undergone should be available immediately when it is needed.

In most cases it is almost impossible to have only “clean” and high-quality data entered into the information system. According to the research report of The Data Warehousing Institute (TDWI), “25% of critical data within Fortune 1000 companies will continue to be inaccurate through 2007. Poor quality customer data costs U.S. business an estimated $611 billion dollars a year in postage, printing, and staff overhead” [2].

Data cleaning, also known as data cleansing and data scrubbing, is the process of maintaining the quality of data. The data cleaning solution should involve discovering erroneous data records, correcting data and duplicate matching.

Data cleaning has two main applications: Master Data Management[10][12] solutions and data warehouses [10] [11], but is also often used in transactional systems.

### *B.* Data Quality Problems

According to [14], data quality issues may be divided into two main categories: issues regarding data coming from one source and issues regarding data from multiple sources. Both main categories may be further divided into subcategories: data quality issues on the instance and on the record level.

#### *1)* One data source – schema level issues

This type of data issues is caused in most cases by source database design flaws. If the source table does not have a primary key constraint that uses a unique record identifier,

978-83-60810-14-9/08/$25.00 © 2008 IEEE

we may have two records referring to separate real world objects bearing the same identifier. If a domain check constraint is missing, a column may contain values outside a specified range, e.g. more than two values describing sex. If a not-null constraint does not exist, a record may lack a value for a mandatory attribute.

If a source schema does not have reference constraint, we may have records in one table referencing non-existent records in another table.

#### *2)* One data source – record level issues

This type of issues is related to the errors that occur at the point of data entry into the source system. Typical errors involve:

- misspellings – caused either by OCR imperfections, e.g. ‘1’ (digit one) may be replaced with ‘l’ (lower case L), typos, or phonetic errors,
- default values for mandatory attributes,
- inconsistencies (e.g. incorrect ZIP code for a city),
- misfielded values, i.e. correct values but placed in wrong attributes, e.g. country=”Warsaw”,
- duplicate records: more than one record referring the same real world object.

#### *3)* Multiple data sources – schema level issues

Multiple source schema level errors are caused by different data models in each of the source systems. For example, there may be homonyms or synonyms at the attribute level (attributes with the same name, but different meaning or attributes with different names but the same meaning), different levels of normalization, different constraints, or different data types.

#### *4)* Multiple data sources – record level issues

This kind of data quality issues is the cross-product of all the aforementioned issues. In addition, there may be other problems:

- different units of measure for the same attribute (meters vs. inches, $ vs. €, etc.),
- Different constraints/domains; e.g. `Sex= {M, F}` ; `Sex= {0, 1}` , etc.,
- Different levels of aggregation: daily vs. weekly, monthly vs. annually,
- Duplicate records - the same object from the real world may have different representations in different source systems.

### *C.* Data cleaning areas and related work

The following primary areas of data cleaning may be distinguished, namely:

- Duplicate matching: in case of integrating multiple sources it may happen that one or more sources contain records denoting the same real world object. The records may have various degrees of data quality. Therefore, one of the tasks of the data cleaning solution is to identify duplicates and join them into a single record whose data quality would be high. This problem is known as “merge/purge” or record linkage problem. Duplicate matching is also used to discover duplicates on the higher level, e.g. records of people that share the same address. This problem is known as household detection [14]. The research in this area (e.g., [3][14] [15][18] ) is focused on devising methods that are both effective, i.e. result in high number of correct matches and low number of incorrect matches, and efficient, i.e. performing within the time constraints defined in the system requirements.
- Data standardization and correction: in case of different domains used in different source systems the data cleaning solution should transform all the values used in those system into one correct set of values used in the target system. Moreover, if any incorrect values appear, the role of the data cleaning is to identify those values and alter them. Works that concern this issue use various methods ranging from statistical data cleaning[8] to machine learning[4][12].
- Schema translation: as source systems may utilize different data models, the task of data cleaning solution is to provide a mapping from those data models to the target data model. This may require splitting free-form fields (e.g. “address line 1”) into an atomic attribute set (“street”,” home no”, “zip code”). The research in this field focuses on devising methods that are capable of performing this process automatically [6]

## II. Application of Data Mining Methods in Data Cleaning

All current data cleaning solutions are highly dependent on human input. The deliverable of the profiling phase - the first phase of a data quality assessment [13], is a set of metadata describing the source data which is then used as an input for the creation of data validation and transformation rules. However, the validation rules have to be confirmed or designed by a business user who is an expert in the business area being assessed. It is not always easy or straightforward to create such a set of business rules. The situation is very similar where duplicate matching is concerned. Even if business rules for record matching are provided, e.g. “Equal SS-N’s and dates of birth”, it may be impossible to match duplicate records, as any of the data quality issues may occur thus preventing from exact matching. Therefore, if incorrect SS-N’s or dates of birth stored in different positional systems occur, exact-matching business rules may not mark the records as duplicates. As far as attribute standardization and correction is concerned, data cleaning solutions are only as good as the reference data they use. Reference data is a set of values which are considered to be valid for a given attribute, e.g. list of states, countries,car models, etc.

The aim of research led by the author of this article was to examine if data mining methods may be used to solve tasks mentioned in the above paragraph. As data mining is the process of discovering previously unknown knowledge from data, it can be used to discover data validation rules and reference data directly from the data set. Table I contains possible application of data mining methods within the area of data cleaning..

The main focus of this article is attribute correction. Other data cleaning tasks are subject to further research. This article presents two applications of data mining techniques in the area of attribute correction: context-independent attribute correction implemented using clustering techniques and context-dependent attribute correction using associations. The next sections of this chapter present two algorithms created by the author that are supposed to examine the possibility of application of data mining techniques in data cleaning. Both algorithms are intended for use in the application of text-based address and personnel data cleaning.

### A. Context independent attribute correction

As mentioned in the previous section, attribute correction solutions require reference data in order to provide satisfying results. The algorithm described in this section is used to examine if a data set may be a source of reference data that could be used to identify incorrect entries and enable to correct them. The algorithm aims at correcting text attributes in address and personal data.

Context-independent attribute correction means that all the record attributes are examined and cleaned in isolation without regard to values of other attributes of a given record.

The key idea of the algorithm is based on an observation that in most data sets there is a certain number of values having a large number of occurrences within the data set and a very large number of attributes with a very low number of occurrences. Therefore, the most-representative values may be the source of reference data. The values with low number of occurrences are noise or misspelled instances of the reference data. Table II shows an excerpt from the distribution with values sorted alphabetically and then descending according to the number of occurrences. The attribute being examined comes from the sample data set used in this experiment and is derived from an Internet survey. As it may be noticed, there are many (6160) records that have the value “Warszawa” and very few that have a value that closely resembles “Warszawa”. Therefore, the value “Warszawa” is most likely the correct value of the attribute and should enter the reference data set while the remaining values are misspelled and should be corrected to “Warszawa”.

The aim of the algorithm is to create a reference data set and a set of substitution rules that could be used to correct misspelled values.

Table II .
Location attribute distribution

| Location | Number of occurrences |
|---|---|
| Warszawa | 6160 |
| Warszwa | 8 |
| Warszaw | 4 |
| Warsawa | 1 |
| Warszawaa | 1 |
| Warzawa | 1 |
| Warzsawa | 1 |

#### 1) Algorithm definition

The algorithm utilizes two data mining techniques: clustering and classification using the “k-nearest neighbours” method.

The algorithm has two parameters: distance threshold - *distThresh* being the minimum distance between two values allowing them to be marked as similar, and occurrence relation – *occRel*, used to determine whether both compared values belong to the reference data set. This parameter was introduced because it is possible that there are two values that may seem similar terms of the distance but that may represent two different objects of the real world that should create separate entries in the reference data set.

To measure the distance between two values a modified Levenshtein distance [5] was used. Levenshtein distance for

Table I.
Location attribute distribution

| | Clustering | Association rules | Classification |
|---|---|---|---|
| **Data standardization/attribute correction** | X | X | |
| **Duplicate matching** | X | | |
| **Validation rules generation** | | | X |
| **Outlier detection** | X | | |
| **Missing attribute prediction** | | X | X |

two strings is the number of text edit operations (insertion, deletion, exchange) needed to transform one string into another.

The algorithm for attribute correction described here utilizes a modified Levenshtein distance defined as

$$\hat{Lev}(s_{1,}s_2)=\frac{1}{2}\cdot(\frac{Lev(s_{1,}s_2)}{\|s_1\|}+\frac{Lev(s_{1,}s_2)}{\|s_2\|}) \quad (1)$$

where $Lev(s_1,s_2)$ denotes a Levenshtein distance between strings $s_1$ and $s_2$. The modified Levenshtein distance for two strings may be interpreted as an average fraction of one string that has to be modified to be transformed into the other. For instance, the Levenshtein distance between “Warszawa” and “Warzsawa” is 2, while the modified Levenshtein distance for “Warszawa” and “Warzsawa” is 0.25. The modification was introduced to be independent of the string length during the comparison.

The algorithm consists of following steps:

1. Preliminary cleaning – all attributes are transformed into upper case and all the non-alphanumeric characters are removed
2. The number of occurrences for all the values of the cleaned data set is calculated.
3. Each values is assigned to a separate cluster. The cluster element having the highest number of occurrences is denoted as the cluster representative.
4. The cluster list is sorted descending according to the number of occurrences for each cluster representative.
5. Starting with first cluster, each cluster is compared with the other clusters from the list in the order defined by the number of cluster representative occurrences. The distance between two clusters is defined as the modified Levenshtein distance between cluster representatives
6. If the distance is lower than the *distThresh* parameter and the ratio of occurrences of cluster representative is greater or equal the *occRel* parameter, the clusters are merged.
7. If all the clustered pairs are compared, the clusters are examined whether they contain values having distance between them and the cluster representative above the threshold value. If so, they are removed from the cluster and added to the cluster list as separate clusters.
8. Steps 4-7 are repeated until there are no changes in the cluster list i.e. no clusters are merged and no clusters are created.
9. The cluster representatives form the reference data set, and the clusters define transformation rules – for a given cluster cluster elements values should be replaced with the value of the cluster representative.

As far as the reference dictionary is concerned, it may happen that it will contain values where the number of occurrences is very small. These values may be marked as noise and trimmed in order to preserve the compactness of the dictionary.

*2) Results*

The algorithm was tested using a sample data set derived from an Internet survey. The data record is define as a tuple {*First Name*, *Last Name*, *Occupation*, *Location*}. There were about 30057 records divided into 6 batches of 5 thousand records. The attribute that was the source of the data for cleaning was “Location”. During review 1166 elements – 3.88% of the whole set, were identified as incorrect and hence subject to alteration.

Table III contains example transformation rules discovered during the execution of the algorithm

Table III.
Example transformation rules

| Original value | Correct value |
|---|---|
| Warszwa | Warszawa |
| Warszaw | Warszawa |
| Warsawa | Warszawa |
| Warzsawa | Warszawa |

In order to test the correctness of the algorithm, following measures are used:

- $p_c$ – percentage of correctly altered values
- $p_i$ – percentage of incorrectly altered values
- $p_0$ – percentage of values marked during the review as incorrect, but not altered during the cleaning process.

The measures are defined as

$$p_c=\frac{n_c}{n_a}\cdot 100$$
$$p_i=\frac{n_i}{n_a}\cdot 100 \quad (2)$$
$$p_0=\frac{n_{00}}{n_0}\cdot 100$$

where $n_c$ is the number of correctly altered values, $n_i$ the number of incorrectly altered values, $n_a$ the total number of altered values, $n_{00}$ the number of elements initially marked as incorrect that were not altered during the cleaning process and $n_0$ the number of values identified as incorrect.

Fig 1 and Table IV show the dependency between the *distThresh* threshold parameter and defined measures.

It can be observed that the percentage of values altered is growing with the increase of the *distThresh* parameter. It is caused by decreasing the restrictiveness of the algorithm in assigning values as similar. The percentage of correctly altered values reaches its highest values of 92.63% at 0.1, however, for that value of the parameter only about 7% of previously identified incorrect entries are corrected. The percentage of incorrectly altered values is also growing due to the greater tolerance for duplicate matching criteria. However, it is possible to define a value of the parameter to achieve the optimal ratio of the correctly altered values avoiding a large number of incorrect values. In the case of the data examined the optimal value of the attribute was 0.2 where 20% of incorrect entries were altered, 79.52% of which were altered correctly. What is still a subject to further experiments is the optimal value of the *occRel* parameter. Also the clustering method used here could be replaced by other clustering methods which could improve the number of correctly altered values, but might have negative influence on the cleaning execution time, as method used here does not require comparison between all the values from the cleaned data set.

The algorithms displays better performance for long strings as short strings would require higher value of the parameter to discover a correct reference value. However, as it was noted in the previous paragraphs, high values of the *distThresh* parameter results in larger number of incorrectly altered elements.

This method produces as 92% of correctly altered elements which is an acceptable value. The range of the applications of this method is limited to elements that can be standardized for which reference data may exist. Conversely, using this method for cleaning last names could end with a failure.

The major drawback of this method is that may classify as incorrect a value that is correct in context of other attributes of this record, but does not have enough occurrences within the cleaned data set.

Table IV.
Dependency between the measures and the *distThresh* parameter for context-independent algorithm

| **distThresh** | $p_c$ | $p_i$ | $p_0$ |
|---|---|---|---|
| 0 | 0.0 | 0.0 | 100.0 |
| 0.1 | 92.63 | 7.37 | 92.45 |
| 0.2 | 79.52 | 20.48 | 36.96 |
| 0.3 | 67.56 | 32.44 | 29.25 |
| 0.4 | 47.23 | 52.77 | 26.93 |
| 0.5 | 29.34 | 70.66 | 23.41 |
| 0.6 | 17.36 | 82.64 | 19.04 |
| 0.7 | 7.96 | 92.04 | 8.92 |
| 0.8 | 4.17 | 95.83 | 1.11 |
| 0.9 | 1.17 | 98.83 | 0.94 |
| 1.0 | 0.78 | 99.22 | 0 |

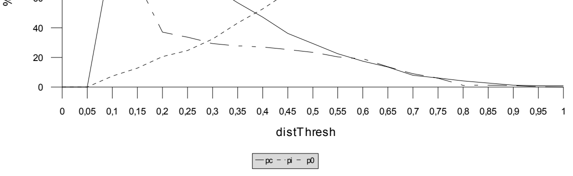


Fig 1: Dependency between the measures and the *distThresh* parameter for context-independent algorithm.

### B. *Context-dependent attribute correction*

The context-dependent attribute correction algorithm is the second of the algorithms designed by the author that utilizes data mining methods. Context-dependent means that attribute values are corrected with regard not only to the reference data value it is most similar to, but also takes into consideration values of other attributes within a given record. The idea of the algorithm is based on assumption that within the data itself there are relationships and correlations that can be used as validation checks. The algorithm generates association rules from the dataset and uses them as a source of validity constraints and reference data.

#### *1) Algorithm definition*

The algorithm uses association rules methodology to discover validation rules for the data set. To generate frequent itemsets the Apriori[17] algorithm is utilized.

The algorithm described in this chapter has two parameters *minSup* and *distThresh*. The first of the parameters - *minSup*, is defined analogically to the parameter of the same name for the Apriori algorithm used here. The other parameter – *distThresh*, is the minimum distance between the value of the "suspicious" attribute and the proposed value being a successor of a rule it violates in order to make a correction.

Although association rules algorithms normally have two parameters: *minSup* and *minConf*, i.e. minimal confidence for generated rules, the algorithm used here does not use the latter of the two. However, the algorithm can be modified to complete the missing attribute values. In such cases the *minConf* can be used to determine the minimal confidence the rule must have in order to fill in the missing value.

For calculating distances between textual attributes the modified Levenshtein distance described in previous section is used.

The algorithm has following steps:

1. Generate all the frequent sets, 2-sets, 3-sets and 4-sets.
2. Generate all the association rules from the sets generated in the previous step. The rules generated may have 1, 2, or 3 predecessors and only one successor. The association rules generated form the set of validation rules.
3. The algorithm discovers records whose attribute values are the predecessors of the rules generated with an attribute whose value is different from the successor of a given rule. These records are marked "suspicious".
4. The value of the attribute for a "suspicious" row is compared to all the successors of all the rules it violates. If the relative Levenshtein distance is lower than the distance threshold, the value may be corrected. If there are more values within the acceptable range of the parameter, a value most similar to the value of the record is chosen.

### 2) *Results*

The algorithm was run on a set of 287198 address records. The data records are tuples defined as `{street, location, zip code, county, state}`.

The rule-generation part of the algorithm was performed on the whole data set. The attribute correction part was performed on a random sample of the dataset consisting of 2851 records. During a review 399 attribute values were identified as incorrect for this set of records .

To verify the performance of the algorithm, measures $p_c$ – the percentage of correctly altered values, $p_i$ – the percentage of incorrectly altered values, and $p_0$ – the percentage of values not altered, as defined in previous section are used.

Table V and Fig 2 show the relationship between the measures and the *distThresh* parameter. The *minSup* parameter was arbitrarily set to 10.

Table V.
Dependency between *distThresh* parameters and measures for context dependent algorithm

| **distThresh** | $p_c$ | $p_i$ | $p_0$ |
|---|---|---|---|
| 0 | 0.00 | 0.00 | 100.00 |
| 0.1 | 90 | 10 | 73.68 |
| 0.2 | 68.24 | 31.76 | 46.62 |
| 0.3 | 31.7 | 68.3 | 36.09 |
| 0.4 | 17.26 | 82.74 | 33.83 |
| 0.5 | 11.84 | 88.16 | 31.33 |
| 0.6 | 10.2 | 89.8 | 31.08 |
| 0.7 | 9.38 | 90.62 | 30.33 |
| 0.8 | 8.6 | 91.4 | 28.82 |
| 0.9 | 8.18 | 91.82 | 27.32 |
| 1.0 | 7.77 | 92.23 | 17.79 |

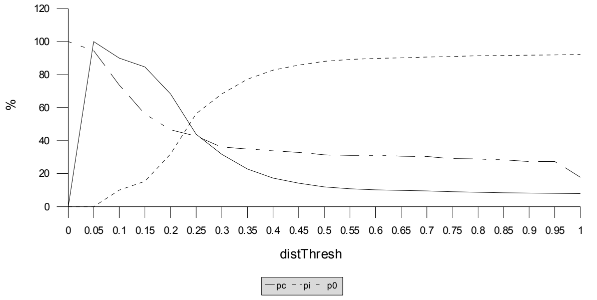


Fig 2: The dependency between the distThresh parameter and measures for context-dependent algorithm.

The results show that the number of values marked as incorrect and altered is growing with the increase of the *distThresh* parameter. Contrary to the same observation in case of context-independent cleaning, this number never reaches 100%. This proves that some attributes that may at first glance seem incorrect, are correct in the context of other attributes within the same record. The percentage of correctly marked entries reaches its peak for the *distThresh* parameter equal to 0.05. The result is better than in the case of context-independent cleaning as the number of correctly altered values for this value of the parameter is equal to the total number of altered values. This also proves that context-dependent cleaning algorithm performs better at identifying incorrect entries. The number of incorrectly altered values is growing with increase of the parameter. However, a value of the *distThresh* parameter can be identified that gives optimal results, i.e. the number of correctly altered values is high and the number of incorrectly altered values is low. In case of this experiment the value of the parameter is 0.15.

Some areas of improvement for this method may be identified. A possible change in the algorithm could involve adding one more parameter – the *minConf* for generated rules. This parameter has the same meaning as the *minConf* parameter of the Apriori algorithm. This would enable pruning the "improbable" rules and limit the number of incorrectly altered values. Also generating the rules using cleaned data would result in better algorithm performance.

## III. Conclusion

The results of the experiments verifying correctness of both algorithms for attribute correction prove that using data mining methods for data cleaning is an area that needs more attention and should be a subject of further research.

Data mining methodologies applied in the area of data cleaning may be useful in situations where no reference data is provided. In such cases this data can be inferred directly from the dataset.

Experimental results of both algorithms created by the author show that attribute correction is possible without an external reference data and can give good results. However, all of the methods described here definitely necessitate more research in order to raise the ratio of correctly identified and cleaned values. As it was discovered in the experiments, the effectiveness of a method depends strongly on its parameters. The optimal parameters discovered here may give optimal results only for the data examined and it is very likely that different data sets would need different values of the parameters to achieve a high ratio of correctly cleaned data.

The above experiments utilized only one string matching distance (Levenshtein distance) was used. It is possible that other functions could result in better output and this should be explored in future experiments.

Moreover, further research on application of other data mining techniques in the area of data cleaning is planned.

## References

[1] R. Agrawal, T. Imielinski, A. Swami "Mining Association Rules between sets of Items in Large Databases" in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp.207-216

[2] B. Beal "Bad Data Haunts the Enterprise" in *Search CRM*, http://searchcrm.techtarget.com/news/article/0,289142,sid11_gci965128,00.html

[3] M. Bilenko, R. Mooney "Adaptive Duplicate Detection Using Learnable String Similarity Measures" in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*

[4] T. Churches, P. Christen, K. Lim, J. Xi Zhu "Preparation of name and address data for record linkage using hidden Markov models", *BMC Medical Informatics and Decision Making*, 2, 2002

[5] W. Cohen, P. Ravikumar, S. Fienberg "A Comparison of String Distance Metrics for Name-Matching Tasks" in *Proceedings of the IJCAI-2003*

[6] W. Cohen "Integration of Heterogeneous Databases without Common Domains Using Queries Based Textual Similarity" in *Proceedings of the 1998 ACM SIGMOD international conference on Management of data* pp. 201-212

[7] M. Ester, H. Kriegel, J. Sander, X. Xu "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise" in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*

[8] B. Fung, K. Wang, M. Ester, F. Fraser "Hierarchical Document Clustering" http://www.cs.sfu.ca/~ester/papers/Encyclopedia.pdf

[9] T. Herzog, F. Scheuren, W. Winkler *Data Quality and Record Linkage Techniques* New York: Springer Science+Business Media, 234pp., ISBN: 978-0-387-69502-0

[10] R. Kimball, M. Ross *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* Wiley, John & Sons, Incorporated, 464pp, ISBN-13: 9780471200246

[11] R. Kimball, J. Caserta *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data* Wiley, John & Sons, Incorporated, 525pp, ISBN-13: 9780764567575

[12] M. Lee, T. Ling, W. Low "IntelliClean: A knowledge-based intelligent data cleaner" in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.290-294

[13] A. Maydanchik *Data Quality Assessment* Technics Publications, 336pp, ISBN-13: 9780977140022

[14] E. Rahm, H.Do "Data Cleaning. Problems and Current Approaches" in *IEEE Bulletin of the Technical Committee on Data Engineering,* Vol 23 No. 4, December 2000

[15] P. Ravikumar, W. Cohen "A Hierarchical Graphical Model for Record Linkage" in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*

[16] K. Ward Church "Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text" in *Proceedings of the Second Conference on Applied Natural Language Processing*

[17] J. Webb "Association Rules" in *The Handbook of Data Mining* Nong-Ye(Ed) Lawrence Erlbaum Associates, Inc., ISBN-13: 9780805855630, 724pp , pp 25-39

[18] W. Winkler "Advanced Methods For Record Linkage" in *Proceedings of the Section on Survey Research Methods* , *American Statistical Association* , pp 467-472

[19] W. Winkler "The State of Record Linkage and Current Research Problems" in *Proceedings of the Survey Methods Section*, *Statistical Society of Canada*, pp 73-80

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A rule-based approach to robust granular planning

Sebastian Ernst
Institute of Automatics
AGH University of Science and Technology
al. Mickiewicza 30
30-059 Kraków
Poland
Email: ernst@agh.edu.pl

Antoni Ligęza
Institute of Automatics
AGH University of Science and Technology
al. Mickiewicza 30
30-059 Kraków
Poland
Email: ligeza@agh.edu.pl

***Abstract*—Real-time execution of planned routes often requires re-planning, especially in dynamic, highly unpredictable environments. Ad-hoc re-planning in case of failure of an optimal (suboptimal) plan leads to deterioration of solution quality. Moreover, it becomes costly and often unmanageable under real time constraints. This paper describes an approach of a partitioning scheme for urban route planning problem domain into a hierarchy of subproblems and an algorithm for rule-based top-down route derivation. A hierarchical, rule-based approach is aimed at granular planning for generating robust, multi-variant plans.**

## I. Introduction

ROUTE planning became a classic problem of Artificial Intelligence [1]. Simultaneously, numerous practical planning algorithms were developed and the worked-out techniques found practical applications in complex, realistic domains. There is a very large group of problems which can be solved by an Artificial Intelligence planning techniques and the AI methods are explored in practice. Such problems include mobile robot route planning, production line scheduling, development of automated game playing strategies and, last but not least, vehicle route planning.

There have been developed a number of well-founded plan generation techniques, mostly based on graph-search procedures [2]. A recent handbook on planning provides a comprehensive review of the domain [3]. Practical applications became a part of everyday life [4], [5], [6], [7].

The route planning problem becomes especially nice and interesting results are obtained when the search takes place in regular, sparse graphs, and such a situation takes place during route planning for long distances with use of highways and motor-roads for most of the planned route. On the other hand, efficient planning in a large, congested city, under real time constraints and under highly unpredictable traffic conditions remains quite a challenge even for most sophisticated systems. In fact, even the problem statement should be different here with respect to the classics of planning a route incorporating mostly motorways and inter-city roads. The unpredictability of urban traffic conditions and the relatively dense road network are both factors intrinsic in the urban route planning and making it a difficult task. Modern planning systems often provide unreliable results for such class of hard planning problems, sometimes failing to provide them in time or do not provide them at all.

This paper describes an approach to planning based on combination of plan from components calculated a priori. A set of rules assigned to each such component provides a ready-to-use solution to be selected depending on external conditions and the aimed destination. Instead of a single, optimal plan, a bunch of similar, alternative plans is in use; when execution of one of them fails, another one is activated immediately. A partitioning scheme for urban route planning problem domain into a hierarchy of subproblems and an algorithm for rule-based top-down route derivation are outlined. A hierarchical, rule-based approach is aimed at granular planning for generating robust, multi-variant plans.

In our previous research, we focused on various aspects of urban route planning which can be improved by using methods of artificial intelligence. These include prediction of traffic conditions [8], induction of granularity over plan solutions [9] and adaptation of well-known computer network routing protocols to broadcast partial solutions between plan nodes [10], [11].

The decision to take up the rule-based hierarchical approach to solving urban route planning problems can be best justified by presenting the currently used methods for route planning, their shortcomings and research done to date regarding improvement of those methods.

### A. State of the art

Commercially available computer-based route planning systems date back to the 1980s. At the time, products such as AutoRoute had to cope with harsh limitations of computer resources. The vast development of processing power even in portable personal computers remedied that problem, but the search algorithms and optimality criteria remained mostly unchanged since that time.

Classical planning algorithms are based on informed graph-search methods, such as various modifications of the famous $A^*$ algorithm and its descendants (e.g. $IDA^*$) [1], [3]. The criteria for optimality used by such "classical" route planning methods are, in most cases, limited to:

- finding the shortest route (the so-called *shortest-path algorithms*,

- time-optimal solutions requiring finding the least time-consuming route (where the travel time is calculated using naive criteria based on the type of route, not taking the prevailing traffic conditions into account),
- finding the least expensive route (by adding estimated fuel consumption and road fees/taxes).

The above generic possibilities can be extended by requirement to go through some indicated middle-points, avoid some forbidden points, avoid left-turn, as well as other user preferences.

In general, the worked out planning methods, combined with the processing power of modern computers, are enough for most applications in long-distance route planning, especially when planning is performed *a priori* – before the travel begins. The search graph is usually relatively sparse, and binary congestion warning/road segment exclusion methods provide sufficient means of route customization.

### B. Shortcomings of classical planning methods

The planning process becomes more problematic complex task when the following conditions occur:

- the search domain becomes more dense, and hence the effective branching factor is too high for systematic planning; too many alternative plans are to be considered,
- more complex criteria have to be taken into account in the search process, especially ones influenced by unknown, unpredictable conditions,
- it is necessary to perform re-planning, due to a change of traffic conditions or driver disobedience,
- robustness of the plan becomes of primary importance, i.e. the user wants to have a *reliable* plan working under almost any conditions instead of an optimal one, but fallible under expected traffic conditions.

The aforementioned shortcomings become especially apparent in situations where planning has to be performed in real time, namely when the planning methods are used in real-time mobile navigation devices. Such devices often offer limited processing power, making algorithm efficiency even more crucial.

### C. Previous research

Efficient plan generation under heavy traffic conditions in complex urban environments has not deserved intensive study in the literature [3]. An approach based on pure graph-search methods seems to be of limited usability. A study of use of methods based on *case-based* approach was presented in [12].

Knowing that pre-computed solutions have to be calculated *a priori*, we focused on several methods providing a compromise between combinatorial explosion and calculation time.

Article [9] introduced a new approach to solving planning problems, based on the concept of maintaining a set of alternative solutions to allow quick plan switching without the need of re-planning. A new concept, called solution robustness, was introduced to allow non-standard optimality criteria. A robust solution is one that is unlikely to fail.

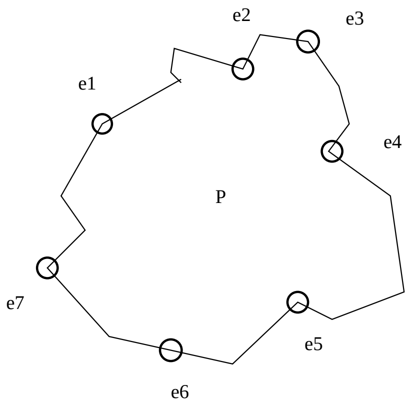


Fig. 1. A schematic presentation of a granule of the planning area with entries.

A different approach was presented in [10]. The concept, based on dynamic computer network routing protocols, is based on partial solution broadcasting between search domain nodes (usually representing junctions). Each node would maintain a “routing table”, providing the next node to be visited in order to reach a given destination node.

## II. Granular hierarchical planning: an intuitive outline of the approach

Since the proposed approach is quite different from the classical ones based on graphs search procedures, below we present an outline of basic ideas of knowledge representation and planning techniques in use.

First, let us point out to the principal assumptions forming foundations of the approach:

- excessive part of planning is performed off-line, a priori, prepared and stored in forms of rules ready-to-use on desire,
- such rules for components of knowledge specific for the area of planning; they can be re-used whenever necessary,
- a plan has hierarchical structure; recursive going up and down is possible,
- instead of a single-thread plan multi-thread (granular ones) are enforced,
- re-planning consist in finding and adapting almost ready plan using prepared a priori knowledge components.

The very basic idea is that a graph modelling the network of routes is divided into separate granules covering it. A granule is and area restricted by some border line where there are some distinguished entry points through which one can enter and leave this area. A good example – for intuition – is a city having several entry roads or a specific district with some entry point. Some illustrative picture of such a granule $P$ with seven entry points is given in Fig. 1. The entry points of granule $P$ are denoted as $P.e1, P.e2, \ldots, P.e7$.

The complete area of planning is divided into such granular components covering it; the principle is to use natural borders and having as small number of entry points as possible. Granules are interconnected by links. In Fig. 2 we have

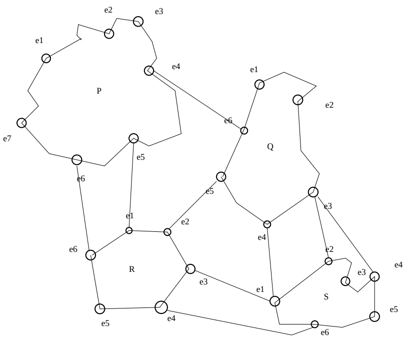


Fig. 2. A schematic presentation of a four interconnected granules within the planning area with entries.

some four interconnected granules, namely $P$, $Q$, $R$ and $S$. For example, granules $P$ and $Q$ are connected by a single link $P.e4--Q.e6$, while granules $Q$ and $S$ are connected through three links, namely: $Q.e4--S.e1$, $Q.e3--S.e2$ and $Q.e3--S.e3$. For simplicity we assume that all such interconnections are symmetric.

Now, a simple plan to go from an initial location $I$ (located within some granule, say $P$) to some destination goal location $G$ (located within some other granule, say S) is a sequence of the form:

$$\begin{aligned}\pi\colon l_I(I--P.e4), P.e4--Q.e6, e6Qe4,&\\ Qe4--S.e1, l_G(S.e1--G),&\end{aligned} \tag{1}$$

where

- $l_I(I--P.e4)$ is a partial plan to move from $I$ to $P.e4$ within $P$; it may be necessary to employ classical search techniques to generate this part,
- $l_G(S.e1--G)$ is a partial plan to move from $S.e1$ to $G$ within $S$; it may be necessary to employ classical search techniques to generate this part,
- $P.e4--Q.e6$, and $Qe4--S.e1$ are just trivial transition links,
- $e6Qe4$ is a set of computed a priori plans to move from $Q.e6$ to $Q.e4$ with $Q$; a plan from the set can be selected according to optimality criteria (a shortest one or a cheapest one) or depending to current time, weather, traffic conditions, etc.

Two further important issues are as follows.

First, any granule can be lowest-level one, where entries are just interconnected through an indivisible network of roads, or a higher level granule covering a set of interconnected granules inside.

Second, for any granule $P$ there is a set of rules, for convenience represented in the tabular XTT form, where each rule is of the form:

$$entry = P.e_i \wedge out = P.e_j \wedge \psi \longrightarrow plan = \pi(P,i,j,\psi).$$

For the granules of the lowest level, a plan $\pi(P,i,j,\psi)$ is just a path through the graph modelling the network of roads. For any higher level granules a plan $\pi(P,i,j,\psi)$ can incorporate higher level notation such as (1); recall that granules of higher levels are recursively divide into lover level ones.

TABLE I
A SCHEME OF A SIMPLE RULE-BASE TABLE FOR STORING PLANS

| entry | out | cond | plan |
|---|---|---|---|
| P.e1 | P.e2 | $\psi_1$ | $\pi(P,1,2,\psi_1)$ |
| P.e1 | P.e3 | $\psi_2$ | $\pi(P,1,3,\psi_2)$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| P.e6 | P.e7 | $\psi_{46656}$ | $\pi(P,6,7,\psi_{46656})$ |

## III. Important definitions

The original form of a route planning problem domain is a graph. Nodes represent individual states (usually junctions), and vertices (edges) represent the connections (roads, streets). Edge weights represent the value of the quality indicator, according to the chosen optimality criteria. As it is possible that there is more than one connection between two nodes, a modified definition for a graph must be used.

*Definition 1 (Graph):* Let $N$ be a finite set of nodes and let $E$ be a finite set of edges. A graph $G$ is a four-tuple

$$G = (N, E, \alpha, \omega) \tag{2}$$

where $\alpha$ and $\omega$ are functions assigning, to every edge $e \in E$, its start node $n_I \in N$ and end node $n_G \in N$, respectively:

$$\alpha : E \rightarrow N, \quad \omega : E \rightarrow N \tag{3}$$

Please note that every node $n \in N$ will have precise $(x, y)$ coordinates, denoting the geographical location of a given junction.

As the graph represents a fragment of a map – i.e. a city or a part of it, it will most likely be connected with the outside world using roads. We shall therefore create virtual nodes at every exit road, representing entry or exit from the current fragment of the map. That set of nodes shall be called terminators and be defined as $N_T \subset N$. Geographic coordinates for such virtual nodes shall be outside the boundaries of the map region.

*Definition 2 (Terminator):* A terminator is a virtually-created node $n \in N_T \subset N$, representing entry into or exit out of the domain of the planning process. As junctions outside the domain are not of interest to the planning process, each node $n \in N_T$ may only have at most *one* edge leading to another (non-virtual) node $n' \notin N_T$, $n` \in N$ and, optionally, at most *one* edge in the reverse direction.

Basing upon the graph definition above, we shall now provide the definitions for hierarchical graph partitioning.

*Definition 3 (Partition):* We shall say that a series of sets $P_1, P_2, \ldots, P_n \subset N$ are partitions and define a partitioning scheme of a graph $G$ if:

- $P_1 \cup P_2 \cup \ldots \cup P_n = E$, i.e. the subgraphs cover all nodes of the graph $G$ being partitioned,
- $P_i \cap P_j = \emptyset$, $i, j = 1, 2, \ldots, n$, $i \neq j$, i.e. any given node can belong to only one of the resulting subgraphs,

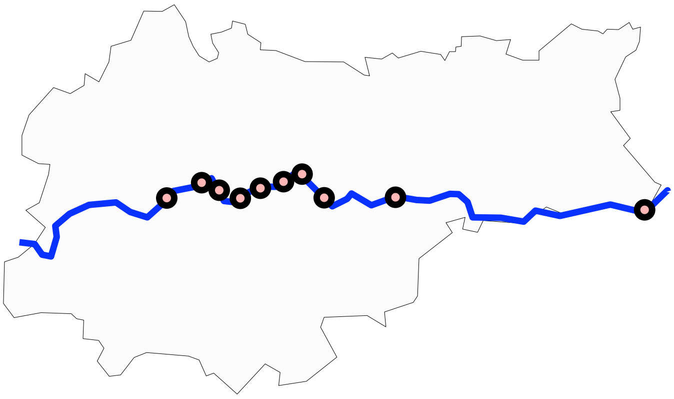


Fig. 3. An outline map of the city of Kraków, showing the boundary between two particles, marked out by the Vistula river, as well as internal links between those partitions.

- every subgraph defined by partitions $P_i$, $i = 1, 2, \ldots, n$ is connected.[1]

In practice, partition boundaries will often be natural or human-made landmarks which provide a limited number of options of crossing them. Bridges or railways are examples of such landmarks. To clarify the concept of the partition, Fig. 3 shows an outline map of the city of Kraków divided naturally into two partitions by the river Vistula. Circles denote the limited number of ways to get from one partition to another. Definitions of neighboring partitions, partition links and partition boundaries are intuitive. Formal definitions follow.

*Definition 4 (Neighboring partitions):* We say that partitions $P_i$ and $P_j$ are neighboring if there is at least one pair of nodes connected directly by an edge such that one node belongs to $P_i$ and the other belongs to $P_j$:

$$\exists n_i, n_j : n_i \in P_i, n_j \in P_j, (n_i, n_j) \in E \tag{4}$$

*Definition 5 (Partition link):* An internal partition link (partition link) between two partitions $P_i$ and $P_j$ is an edge $l \in E$, $l = (n_i, n_j)$, such that $n_i \in P_i$ and $n_j \in P_j$. We say that link $l$ is associated with partitions $P_i$ and $P_j$: $l \sim P_i$, $l \sim P_j$.

Intuitively, a partition link provides the means of getting from one partition to another. Coordinates of a link can in fact be defined as an arbitrarily chosen point belonging to the road leading from junction $n_i$ to junction $n_j$. In practice, this will often be the crossing point between the road and the landmark object used to define the partition in the first place.

On Figure 3, internal links (in this case: bridges) are represented as small circles.

*Definition 6 (External partition link):* Let $G$ be a graph being the input for the partitioning process at level $m$, as described by section III-A. An external partition link is an edge $l_x \in E$, $l_x = (n_i, n_j)$ such that:

- $n_i \in N_T$ or $n_j \in N_T$, or:
- for $m = 1, 2, 3, \ldots$ $(m \neq 0)$, $n_i$ and $n_j$ belong to two neighboring partitions at level $m - 1$.

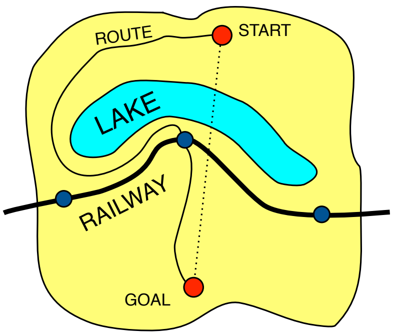


Fig. 4. An example map with a missing partition boundary and the resulting non-optimal solution.

*Definition 7 (Partition boundary):* The boundary $L_i$ of partition $P_i$ is defined by the set of all its links associated with it:

$$L_i = \{l \in E : l \sim P_i\} \tag{5}$$

Geometric representation of the boundary can be perceived as a series of line segments, joining successive pairs of links, in a way that each linking road is crossed only once.

On Figure 3, the Vistula river is the boundary.

Every partition shall have an array of traversal rules.

*Definition 8 (Traversal rule):* Let $P$ be a partition as defined by Definition 3, and $L$ be its boundary, as defined by (5). A traversal rule is a five-tuple:

$$R = (P, l_I, l_G, g, \Phi, \Psi) \tag{6}$$

where $P$ is the partition, $l_I \in E$ is the entering link, $l_G \in E$ is the exit link, $g$ is the traversal cost (depending on chosen optimality criteria), $\Phi$ is an optional set of optional validity values (e.g. time of day) and $\Psi$ is the traversal plan to be followed.

*Definition 9 (Traversal plan):* A traversal plan $\Psi$ of length $n - 1$ from link $l_I$ to link $l_G$ through partition $P$ at level $m$ is defined as follows:

$$\Psi = (l_I, P_1, l_1, P_2, l_2, \ldots, l_{n-1}, P_n, l_G) \tag{7}$$

where $P_i$ $(i = 1, 2, \ldots, m)$ are partitions at level $m + 1$ (sub-partitions of $P$), and $l_i$ $(i = 1, 2, \ldots, m)$ are links between partitions $P_i$ and $P_{i+1}$.

### *A. Partition hierarchy*

As mentioned before, the partitioning scheme is hierarchical, e.g. every partition $P_i$ on level $n$ may become the source graph for partitioning on level $n + 1$. On level 0, the source graph is identical to the search domain.

*Definition 10 (Elementary partition):* A partition $P_i$ at level $n$ which is not divided into other sub-partitions (i.e., no sub-partitions of it exist at level $n + 1$) shall be called an *elementary partition*.

Partitioning should be performed to the point when route planning within a partition can be performed in negligible time using classical shortest path algorithms.

[1] A graph is connected if there is a path connecting any two nodes.

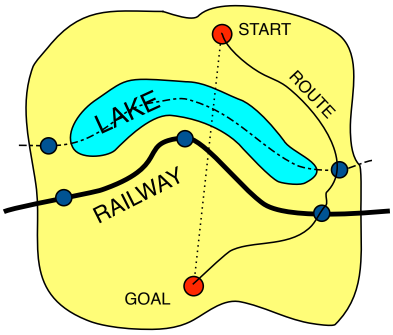


Fig. 5. An example map after adding the missing partition.

In order to maintain consistency throughout the entire hierarchy, partitioning has to be balanced.

*Definition 11 (Partition balancing):* If $L_i$ is the boundary of partition $P_i$ at level $n$, and that partition becomes the source for partitioning at level $n+1$, the set of external links $L_x$ for partitions created by dividing partition $P_i$ equals the boundary of that partition ($L_i$) at level $n$.

## IV. Map partitioning procedures

The process of map partitioning is where intelligent processing actually takes place, thus it requires expert knowledge to be performed properly – to ensure that the entire system yields feasible and optimal results.

The complexity level of the problem makes development of automatic partitioning methods difficult. Therefore, we shall now focus on providing general guidelines for map partitioning, and to discuss the implementation feasibility of an automatic partitioning solution.

### A. Guidelines for partitioning strategies

The primary guideline for partitioning the map is appropriate selection of abstraction levels and partition boundaries. For instance, if a significant landmark is likely to be an obstacle, a partition boundary should be placed accordingly to its size.

For example, please compare the partitioning schemes presented on figures 4 and 5. On figure 4, the partition has been split into two sub-partitions (at the next level), using the railway as the boundary. The algorithm presented below uses geometric criteria to order (or choose) the links used (dotted line between the start and the goal). Therefore, it will assume that the middle railway crossing is the one most likely to be used. If the planning strategy is unfavorable (i.e., full review is not done)

### B. Traversal plan calculation

For each non-elementary partition, a traversal plan needs to be established. Similarly to the partitioning itself, this article assumes that the traversal plans are prepared by experts. However, rough plans may be pre-computed using simple criteria, such as location of exits and known road network irregularities. Such aid should provide satisfactory results, especially on high abstraction levels. However, it should in all cases be reviewed by an expert.

Automated map partitioning and traversal plan calculation methods are indeed possible, but they are not covered in this article, as they involve preparation of the search domain, and not solving the planning problem itself.

## V. Algorithm proposal

Please note that the algorithm is in an early stage of development and therefore some of the assumptions made by it can seem rather naive. Its purpose is to illustrate which stages benefit from using the foundations described in the preceding section.

The algorithm is divided into two main procedures:

- calculate point-to-point route,
- traverse entire partition.

The first procedure (described in section V-A) does not take great benefits from the approach described here, but is needed as an "access road" to true partition traversal. Please note that at any level, the point-to-procedure does not utilize the rule tables associated with partitions.

The *Traverse entire partition* procedure (described in section V-B) is where rule-based processing takes place. No search procedures are utilized, until the elementary partitions are reached and the subproblems become trivial.[2]

### A. Calculate point-to-point route

This is the main procedure used for route planning.

1) Input arguments: $n_I$ (start node), $n_G$ (goal node).
2) Recursively locate $n_I$ and $n_G$ to partitions on all applicable levels. (Determine start and goal node addresses).
3) Assume that $P_I$ is the level $0$ partition of node $n_I$, and $P_G$ is the level $0$ partition of node $n_G$.
4) Define point $p$. Assign the $n_I$ to $p$.
5) Allocate a node of the search graph used to store search options. Assign $n_I$ to the node.
6) Repeat: (build the search tree)
    a) Draw a line segment from $p$ to $n_G$ and select the internal link $l \sim P_I$.
    b) Add the links between $P_I$ and its neighboring partitions as nodes of the search graph, ordered by the distance from the line drawn in the previous point[3], and add appropriate edges.
7) For all nodes linked with the start node $n_I$, recursively calculate the point-to-point route from $n_I$ to the geographic location of the link defined by the given node.
8) For all nodes linked with the goal node $n_G$, recursively calculate the point-to-point route from the geographic location of the link defined by the given node to $n_G$.

[2] Nevertheless, routes for elementary partitions can also be pre-calculated in order to achieve full independence from search algorithms in the *Traverse entire partition* part of the algorithm.

[3] To reduce complexity, the number of nodes added at each level can be limited, either by setting a limit or by defining the maximum distance from the line drawn in step 6a.

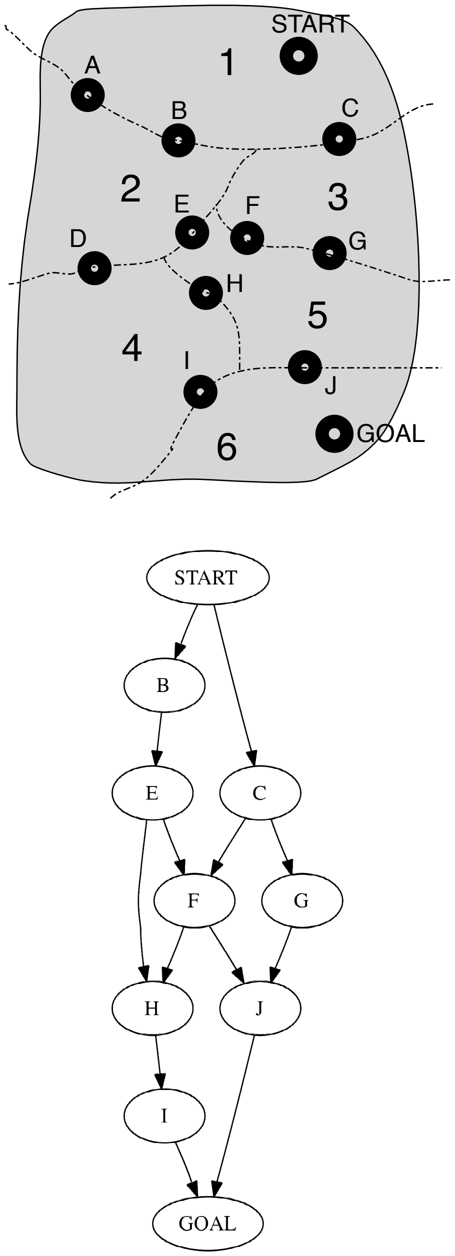


Fig. 6. An example division of a planning problem in to subproblems.

9) For all other nodes, solve the Traverse entire partition problem from their parents in the search tree.
10) For all leaves of the search tree, calculate the point-to-point route from the geographic location of the link defined by the given node to $n_G$.

### B. Traverse entire partition

This procedure is executed if there is an entire partition (at a given level) to be traversed. It only applies to non-elementary partitions (see definition 10).

1) Input arguments: $P$ (partition), $l_I$ (start link), $l_G$ (goal link), $\Phi$ (additional criteria; optional).
2) Match rule $R = (P, l_I, l_G, g, \Phi, \Psi)$ with the most favorable value of $g$.
3) Retrieve traversal plan $\Psi$ ($\Psi = (l_0, P_1, l_1, P_2, l_2, \ldots, l_{n-1}, P_n, l_n)$).
4) Let $n$ be the length of $\Psi$.
5) For $i$ from 1 to $n$, recursively traverse entire partition $P_i$ from node $l_{i-1}$ to node $l_i$.

### C. Execution example

As the *Calculate point-to-point route* algorithm may not seem trivial, we provide an example of its operation. Figure 6 shows a rather simple partition, divided into 6 sub-partitions, labelled 1-6. Internal links between those partitions have been labelled as A-J. The start node is located in partition 1, the goal node – in partition 6. Below the drawing is the search graph generated by the procedure described in section V-A.

The *Calculate point-to-point route* algorithm will be used to solve the following subproblems:

- $START \rightarrow B$
- $START \rightarrow C$
- $I \rightarrow GOAL$
- $J \rightarrow GOAL$

The *Traverse entire partition* algorithm will be used to solve the following subproblems:

- $B \rightarrow E$
- $C \rightarrow F$, $C \rightarrow G$
- $E \rightarrow F$
- $F \rightarrow H$, $F \rightarrow J$
- $G \rightarrow J$
- $H \rightarrow I$

## VI. Conclusion

The article provides some outline for a new rule-based approach to solving planning problems, with special emphasis on vehicle route planning in dense, urban areas under unpredictable traffic conditions. In such cases re-planning may be of excessive use, and granular planning may proof to constitute a robust approach.

The algorithm provided is at an early stage of development, but is a good illustration of the formal foundations described above.

Future work includes:

- development of automatic map partitioning schemes,
- practical implementation of described methods,
- improvement of the search algorithms.

## References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[2] N. J. Nilsson, *Principles of Artificial Intelligence*. Tioga Publishing Co., 1980.

[3] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004, iSBN-10:1-55860-856-7.

[4] Wikipedia, "Google maps," http://pl.wikipedia.org/wiki/Google_Maps, 2007.

[5] ——, "Zumi," http://pl.wikipedia.org/wiki/Zumi, 2007.

[6] Map24, "Map24.interia.pl," http://map24.interia.pl/, 2007.

[7] ViaMichelin, "Route planning systems," http://www.viamichelin.com/.

[8] S. Ernst and A. Ligęza, “Analiza mot’liwoąci zastosowania metod int’ynierii wiedzy do budowy inteligentnego systemu planowania trasy w ruchu miejskim,” in *Int’ynieria wiedzy i systemy ekspertowe*, vol. 2, WrocŞaw, Poland, 2006, pp. 25–34.

[9] ——, “Adaptive granular planning for robust plan generation under uncertain traffic conditions,” in *Proceedings of the 16th international conference on Systems Science*, vol. 2, WrocŞaw, Poland, 2007, pp. 388–396.

[10] ——, “Solving planning problems by broadcasting partial solutions,” in *Tools of information technology*, A. Kos, Ed., Rzeszów, Poland, 2007, pp. 79–84.

[11] S. Ernst, “A multi-agent implementation of an automated planner for dynamic environments,” in *Computer Methods and Systems*, M. S. Ryszard Tadeusiewicz, Antoni LigŚza, Ed., Kraków, Poland, 2007, pp. 99–104.

[12] K. Z. Haigh, J. R. Shewchuk, and M. M. Veloso, “Exploiting domain geometry in analogical route planning,” Journal of Experimental and Theoretical Artificial Intelligence, 1997.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 113–118

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Support Vector Machines with Composite Kernels for NonLinear systems Identification

Amina El Gonnouni, Abdelouahid Lyhyaoui
Engineering System
Lab.(LIS)
Abdelmalek Essaidi University
Tangier, Morocco
Email: amina_elgo@yahoo.fr, lyhyaoui@gmail.com

Soufiane El Jelali, Manel Martínez Ramón
Departamento de Teoria de
la Señal et Comunicaciones
Universidad Carlos III
De Madrid
Email: {soufiane, manel}@tsc.uc3m.es

***Abstract*—In this paper, a nonlinear system identification based on support vector machines (SVM) has been addressed. A family of SVM-ARMA models is presented in order to integrate the input and the output in the reproducing kernel Hilbert space (RKHS). The performances of the different SVM-ARMA formulations for system identification are illustrated with two systems and compared with the Least Square method.**

## I. Introduction

SYSTEM identification treats the problem of constructing mathematical models from observed input and output data. Three basic entities must be taken into consideration to construct a model from data [1]:

1) Data: represent the input and the output data of the system;
2) Candidate models: are obtained by specifying within which collection of models the suitable one exists;
3) Identification method: determining the best model guided by the data.

In the literature of system identification, a large variety of nonlinear methods were used, such as neural networks, high order statistic and fuzzy system [2], [3], [4]. However these models have weaknesses. For example in neural network case, some problems appear, like slow convergence speed and local minima. Support Vector Machines (SVMs) overcomes these problems and seems to be a powerful technique for nonlinear systems where the required model complexity is difficult to estimate.

The Support Vector Machines (SVM) was originally proposed as an efficient method for pattern recognition and classification [3]. Then the technique became a general learning theory. The Support Vector regressor (SVR) was subsequently proposed as the SVM implementation for regression and function approximation [5]. SVM has been widely used to solve problems in text recognition, bioinformatics [6] and bioengineering or image processing [7] and these represent only a few of the practical applications of support vector machines. The key characteristic of SVM is that it maps the input space into a high dimensional feature space or a reproducing kernel Hilbert space through some nonlinear mapping, chosen a priori, in which the data can be separated by a linear function.

The autoregressive and moving average (ARMA) modelling is used when the candidate model is linear and time invariant. The explicit consideration of ARMA models in some reproducing kernel Hilbert space (RKHS) based on support vector machines (SVM-ARMA$_{2k}$) presents a new approach for identifications applications [9]. An analytical relationship between residuals and SVM-ARMA coefficients allows the linking of the fundamentals of SVM with several classical system identification methods. Additionally the effect of outliers can be cancelled [9]. By using the Mercer's Kernels trick, a general class of SVM-based nonlinear system identification can improve model flexibility by emphasizing the input-output cross information (SVM-ARMA$_{4k}$), which leads to straightforward and natural combinations of implicit and explicit ARMA models (SVR-ARMA$_{2k}$) and SVR-ARMA$_{4k}$) [10].

In this paper, we present the different SVM-ARMA models for the system used in [9] and for Bessel difference equation. We present the sensitivity of the SVM-ARMA models to the training data and to the noise power. Additionally, we compare our models with the least square method (LS) and we show each one's performance and the moment they exhibit the same results.

This work is structured as follows: we present the SVR algorithm for nonlinear system identification in section I. In section II, we summarize the explicit ARMA models in RKHS. Simulations and examples are included in section IV. Finally, in section V, we conclude the work.

## II. SVR System Identification

Consider a nonlinear system whose input and output are DTP $\{x_n\}$ and $\{y_n\}$. Let $u_n = [u_n, u_{n-1}, \dots, u_{n-Q+1}]$ and $y_{n-1} = [y_{n-1}, y_{n-2}, \dots, y_{n-P}]$ represent the states of input and output DTP at instant n. The vector $z_n = [y_{n-1}^T, u_n^T]^T$ correspond to the concatenation of the two DTP at that instant *n*.

Giving a training set $\{z_i, y_i\}_{i=1}^N \in \Re^d$ with $d = P + Q - 1$. The linear regression model is:

$$y_n = \langle w, \phi_n(z_n)\rangle + e_n \; . \tag{1}$$

where $\phi(z_n) : \Re^P \times \Re^Q \to H_z$ represents the high dimensional feature space, or RKHS, which is nonlinearly mapped from the

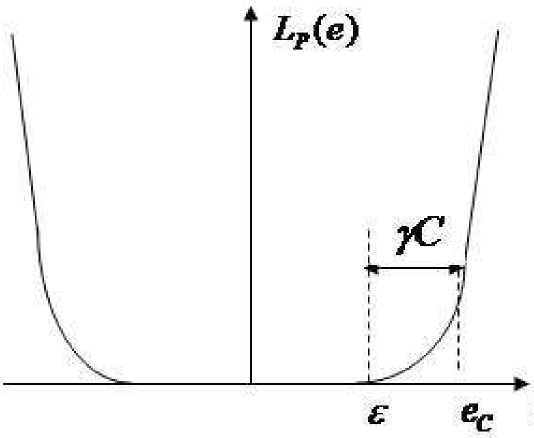


Fig. 1. $\varepsilon$-Huber cost Function.

input space,$\langle\cdot,\cdot\rangle$ represents the dot product and $e_n$ denotes error terms, or residuals, comprehends both measurement and model approximation errors.

In SVR, several cost functions for residuals (CFR) have been used, such as Vapnik's loss function [3], Huber's robust cost [8] or the ridge regression approach [6]. However, in [9] they used $\varepsilon$-Huber CFR, which is a more general cost function that has the above-mentioned ones as particular cases. This cost function is depicted in 1 and it is expressed as:

$$l_p(e_n) = \begin{cases} 0, & |e_n| < \varepsilon \\ \frac{1}{2\gamma}(|e_n| - \varepsilon)^2, & \varepsilon < |e_n| < e_C \\ C(|e_n| - \varepsilon) - \frac{1}{2}\gamma C^2, & |e_n| > e_C \ . \end{cases} \tag{2}$$

where $e_C = \varepsilon + \gamma C$. The $\varepsilon$-Huber CFR can deal with different kinds of noise thanks to the three different intervals.

Using the $\varepsilon$-Huber CFR cost function, the algorithm of SVR system identification corresponds to the minimization of:

$$L_P = \frac{1}{2}\sum_{j=1}^{H_z} w_j^2 + \frac{1}{2\gamma}\sum_{n\in I_1}(\xi_n^2 + \xi_n^{*2}) + C\sum_{n\in I_2}(\xi_n + \xi_n^*) - C\sum_{n\in I_2}\frac{\gamma C^2}{2} \ . \tag{3}$$

with the constraints:

$$y_n - w^T\phi_n(z_n) \le \varepsilon + \xi_n \qquad \forall n = n_0, \cdots, N \ . \tag{4}$$

$$-y_n + w^T\phi_n(z_n) \le \varepsilon + \xi_n^* \qquad \forall n = n_0, \cdots, N \ . \tag{5}$$

where $\xi_n$,$\xi_n^*$ are the slack variables or losses, $\xi_n^{(*)} \ge 0$ ($\xi_n^{(*)}$ represents both $\xi_n$and $\xi_n^*$), $I_1$ is set of samples for which $\varepsilon < \xi_n^{(*)} < e_C$, $I_2$ is the set of samples for which $\xi_n^{(*)} > e_C$, $n_0$ is given by the initial conditions and $N$ is the number of available samples.

By introducing a nonnegative coefficient, Lagrange multiplier, for each constraint ($\alpha_n$ to (4) and $\alpha_n^*$ to (5)), we obtain the Lagrangian for this problem [6] this way:

$$\begin{aligned} L_{PD} = {} & \frac{1}{2}\sum_{j=1}^{H_z} w_j^2 + \frac{1}{2\gamma}\sum_{n\in I_1}(\xi_n^2 + \xi_n^{*2}) + C\sum_{n\in I_2}(\xi_n + \xi_n^*) \\ & - C\sum_{n\in I_2}\frac{\gamma C^2}{2} + \sum_{n=n_0}^{N}\alpha_n(y_n - w^T\phi_z(z_n) - \varepsilon - \xi_n) \\ & + \sum_{n=n_0}^{N}\alpha_n^*(-y_n + w^T\phi_z(z_n) - \varepsilon - \xi_n) \ . \end{aligned} \tag{6}$$

By minimizing the Lagrangian with respect to the primal variables $w_j$ and $\xi_n^{(*)}$ we obtain:

$$w = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)\phi_z(z_n) = \sum_{n=1}^{N}\beta_n\phi_z(z_n) \ . \tag{7}$$

and $0 < \alpha_n^{(*)} < C$, where $\beta = \alpha_n - \alpha_n^*$

The dual problem is obtained by introducing (7) in (6) and it is expressed as:

$$\begin{aligned} L_D = {} & -\frac{1}{2}(\alpha - \alpha^*)^T[G + \gamma I](\alpha - \alpha^*) + (\alpha - \alpha^*)^T y \\ & + \varepsilon 1^T(\alpha - \alpha^*) \ . \end{aligned} \tag{8}$$

where $G$ is gram matrix of dot product or kernel matrix with $G_{ij} = \langle\phi_z(z_i), \phi_z(z_j)\rangle = K_z(z_i, z_j)$, $\alpha_n^{(*)} = [\alpha_1^{(*)}, \ldots, \alpha_N^{(*)}]^T$ and $y = [y_1, \ldots, y_N]^T$. Finally the predicted output for a new observed sample $y_r$ given $z_r$is:

$$\hat{y}_r = \sum_{n=1}^{N}\beta_n K_z(z_n, z_r) \ . \tag{9}$$

With the kernel function $K_z(z_n, z_r)$, we can deal with feature space of arbitrary dimension without having to compute the map $\phi_z$explicitly. Any function that satisfies Mercer's condition can be used as the kernel function [6]. The widely used Gaussian Mercer's kernel is given by $K_z(z_i, z_j) = exp(\frac{-\|z_i - z_j\|^2}{2\sigma^2})$, where $\sigma^2$ is the kernel parameter.

## III. SVR System Identification and Composite Kernels

A family of composite kernels appears in SVM formulation by exploiting the direct sum of Hilbert spaces [10], which allow us to analyse the explicit form of ARMA process in feature space.

### A. Explicit ARMA In Feature Space

By using two possibly different nonlinear mappings $\phi_n(u_n) : \Re^Q \to H_u$ and $\phi_y(y_n) : \Re^P \to H_y$, the input and output state vectors $u_n$ and $y_n$ can be separately mapped to RKHS $H_x$ and $H_y$. So, an ARMA difference equation can be built using two linear models; MA (moving average) in $H_x$ and AR (auto regressive) in $H_y$:

$$y_n = a^T\phi_n(y_{n-1}) + b^T\phi_n(u_n) + e_n \ . \tag{10}$$

where $a = [a_1, \dots, a_{H_u}]^T$ and $b = [b_1, \dots, b_{H_y}]^T$ are vectors representing the coefficients MA and AR of the system, respectively, in RKHS. After formulating the primal problem, stating the Lagrangian and making its gradient to zero, removed the primal variables and formulating the dual problem, the SVM-ARMA$_{2k}$ is obtained by including the kernel matrix $K(z_n, z_r) = K_y(y_{n-1}, y_{r-1}) + K_u(u_n, u_r)$ in (9) [10]:

$$\hat{y}_r = \sum_{n=1}^{N} \beta_n (K_y(y_{n-1}, y_{r-1}) + K_u(u_n, u_r)) \ . \qquad (11)$$

where $K_y(y_{i-1}, y_{j-1}) = \langle \phi_y(y_{i-1}), \phi_y(y_{j-1}) \rangle$ and $K_u(u_i, u_j) = \langle \phi_u(u_i), \phi_u(u_j) \rangle$ are two different Gram matrices, one for the input and the other for the output.

### B. Composite Kernels

The SVM-ARMA$_{2k}$ model could be limited in some cases, because (11) provides an apparent uncoupling between the input and the output. This limitation will be come out explicitly when strong cross information between the two DTP is present. An SVM-ARMA model considering the input and output could simultaneously solve this problem. By using the sum of Hilbert spaces property, the kernel components are:

$$\begin{aligned} K(z_i, z_j) = K_y(y_{i-1}, y_{j-1}) + K_x(u_i, u_j) \\ + K_{xy}(u'_i, y'_{j-1}) + K_{yx}(y'_{i-1}, u'_j) \ . \end{aligned} \qquad (12)$$

When including this kernel in (9), we obtain the SVM-ARMA$_{4k}$ [10].

A new algorithm SVR-ARMA$_{2k}$ can be built by considering the combination between SVR and SVM-ARMA$_{2k}$

$$K(z_i, z_j) = K_y(y_{i-1}, y_{j-1}) + K_x(u_i, u_j) + K_z(z_i, z_j). \quad (13)$$

or an other one, SVR-ARMA$_{4k}$ by combining SVR and :

$$\begin{aligned} K(z_i, z_j) = K_y(y_{i-1}, y_{j-1}) + K_x(u_i, u_j) \\ + K_{xy}(u'_i, y'_{j-1}) + K_{yx}(y'_{i-1}, u'_j) + K_z(z_i, z_j). \end{aligned} \quad (14)$$

## IV. Experimental Results

To examine the performance of SVM-ARMA formulations and to compare it with standard SVR and Least Square method, we use two examples. We focus on radial basis function (RBF) $K_z(z_i, z_j) = exp(\frac{-\|z_i - z_j\|^2}{2\sigma^2})$, where $\sigma^2 \in \Re$ represent the width of the kernel.

For the first example, the prediction performance is evaluated using the mean square error in test set:

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$

where $N$ denotes the total number of data points in the test, $y_i$, $\hat{y}_i$ are the actual value and prediction value respectively

For the second example, we use the normalized mean square error in test set:

$$nMSE = \log_{10} \sqrt{\frac{MSE}{var(y)}}$$

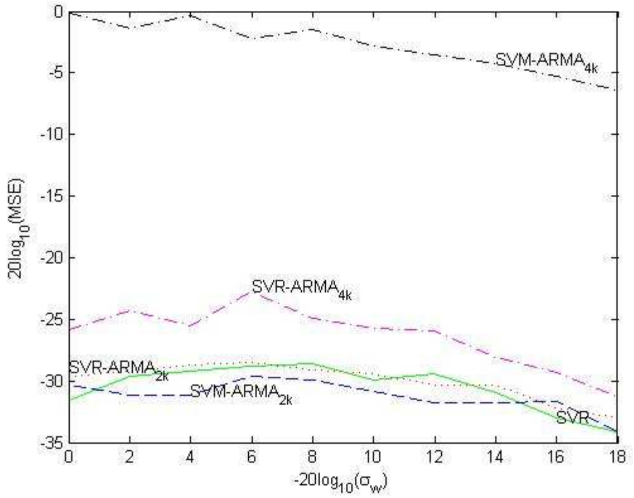


Fig. 2. The MSE as a function of additive noise of power $\sigma_w$.

To train our models, we use the cross validation method; 100 data are used as training data and 100 as testing data. For the first example, the results are averaged over 100 realizations and for example 2, over 200 realizations.

**Example1:** The first example of a system to be identified is [9]:

$$y_n = 0.03y_{n-1} - 0.01y_{n-2} + 3x_n - 0.5x_{n-1} + 0.2x_{n-2} \ . \quad (15)$$

The Input DTP is a white Gaussian noise sequence of unit variance $\{x_n\} \sim N(0, 1)$. An additive small variance random process, $\{e_n\} \sim N(0, 0.1)$, corrupts the corresponding output DTP and modelling the measurement errors. The observed process is $\{o_n\} = \{y_n\} + \{e_n\}$.

Impulsive noise $\{j_n\}$ is generated as a sparse sequence, for which 30% of the samples, randomly placed, are of high-amplitude, having the form $\pm 10 + U(0, 1)$, where $U()$ represents the uniform distribution in the given interval. The remaining are zero samples. The observations consist of DTP input $\{x_n\}$ and the observed output plus impulsive noise; $\{o_n\} + \sigma_w\{j_n\}$. Values of $\sigma_w$ go from 18 to 0 dB [9].

We tried various values for $\varepsilon$, $\gamma$, $C$. For all the SVM-ARMA formulations, $\varepsilon = 0$ is used. In SVM-ARMA$_{4k}$, the values of SVM parameters that give the minimum MSE in testing set are like $C = 1$ and $\gamma = 0.01$, but for other SVM-ARMA models they are fixed in $C = 100$ and $\gamma = 0.001$. The results of our first system are shown in Figure 2, 3, 4, 5, 6. In Figure2, the SVM-ARMA$_{2k}$ model exhibit better performance, whereas SVM-ARMA$_{4k}$ and SVR-ARMA$_{4k}$ provide a poor model in terms of prediction error, that can be explained by the poor cross information between the input and output.

On the other hand, we compare the performance of SVM-ARMA models with the least square method, in which we use the same expression of the kernel components in each case (for example, in the case of SVR-ARMA$_{2k}$, the kernel components are like $_K = K_x + K_y + K_z$ for SVM and LS methods). The results are shown in Fig.3-a, 4-a, 5-a, 6-a, 7-a, and they show

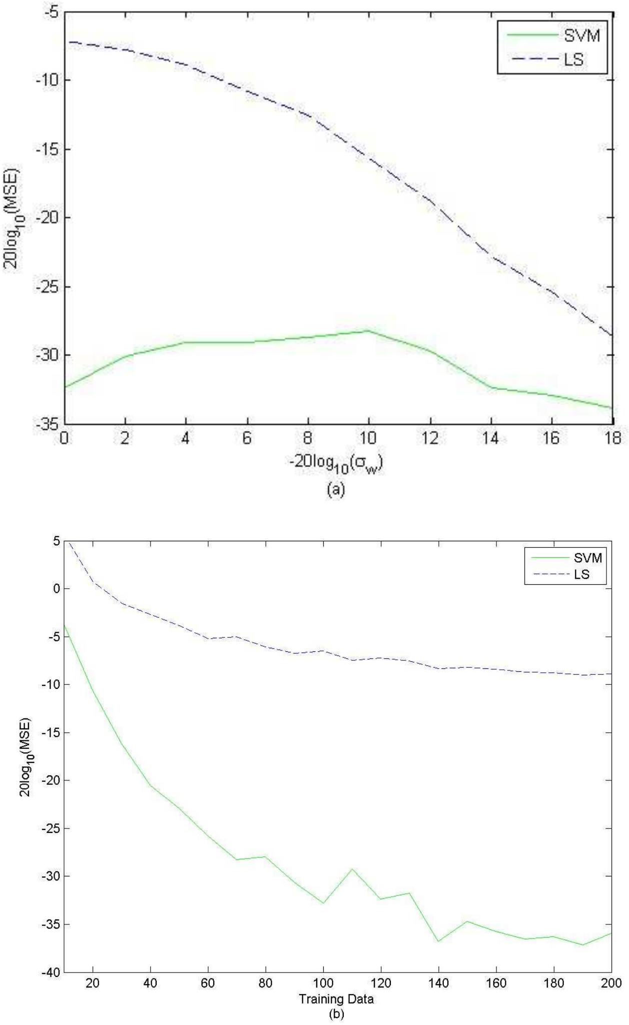


Fig. 3. (a) The MSE as a function of additive noise of power $\sigma_w$ for SVR model. (b) The MSE as a function of training data for SVR model, where $\sigma_w = 1$.

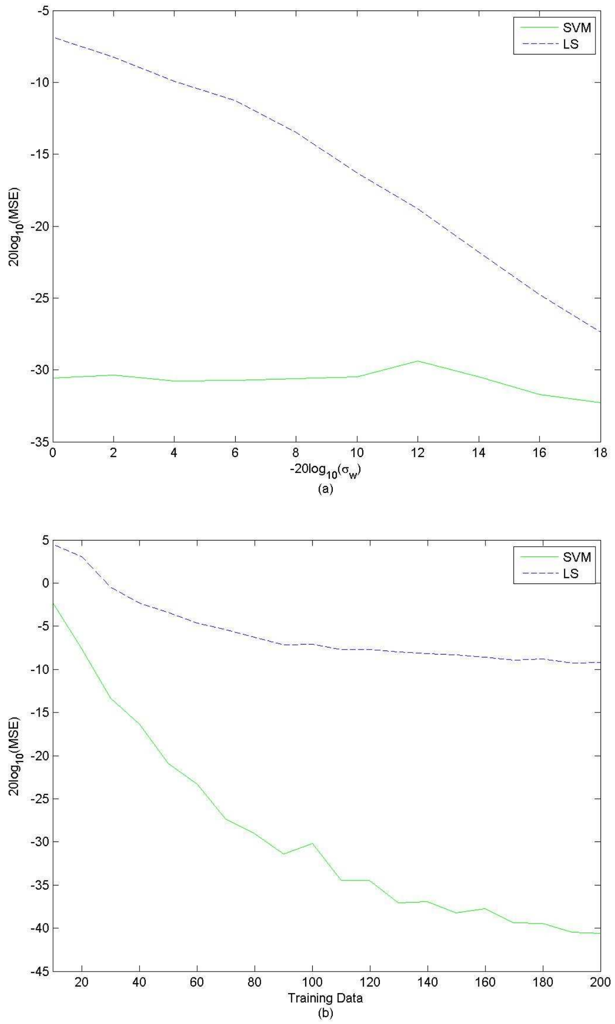


Fig. 4. (a) The MSE as a function of additive noise of power $\sigma_w$ for SVM-ARMA$_{2k}$ model. (b) The MSE as a function of training data for SVM-ARMA$_{2k}$ model, where $\sigma_w = 1$.

that the SVM method exhibits a good performance in high impulsive noise power with a difference of almost 24 dB in comparison with LS method. Besides, SVM-ARMA methods show that there is no significant difference between the different values of MSE as a function of noise parameter, $\sigma_w$, which mean that the SVM-ARMA models, in this example, are not sensitive to the noise parameter $\sigma_w$.

Fig.3-b, 4-b, 5-b, 6-b, 7-b show that in the case of high impulsive noise power, $\sigma_w = 1$, the minimum MSE of SVM and LS methods are stabilized in affixed values even if the number of training data is augmented. We can say that the MSE is saturated. The SVM method needs 160 training data to saturate and LS requirements 100 data, but SVM gives a very small MSE in comparison with LS.

**Example2**: The second system to be identified is described by the difference equation of Bessel:

$$\left\{ \begin{array}{c} u(t) = 0.6\sin^{\alpha}(\pi t) + 0.3\sin(3\pi t) + 0.1\sin(\alpha t) \\ \tilde{y}(t+1) = \frac{\tilde{y}(t)\tilde{y}(t-1)[\tilde{y}(t)-2.5]}{1+\tilde{y}^2(t)+\tilde{y}^2(t-1)} \\ y(t+1) = \Re[\tilde{y}(t+1)] \end{array} \right. . \quad (16)$$

where $\Re[.]$ denotes the real part and $\alpha$ is a random variable uniformly distributed in the interval $[3,4]$ with the mean $E\{\alpha\} = 3.5$.

For all the SVM-ARMA formulations, the SVM parameters, $\varepsilon = 0$, $C = 100$ and $\gamma = 0.01$ are used.

From Table 1, we notice that Bessel equation exhibits the best performance in SVM-ARMA$_{2k}$ and that the SVM-ARMA algorithms show the same results as the LS method.

Table 2 reports the nMSE of Bessel difference equation corrupted with additive Gaussian noise. The SVM formulations give the same value of nMSE, 0.003 dB, and the SVM method exhibits the same results as LS method.

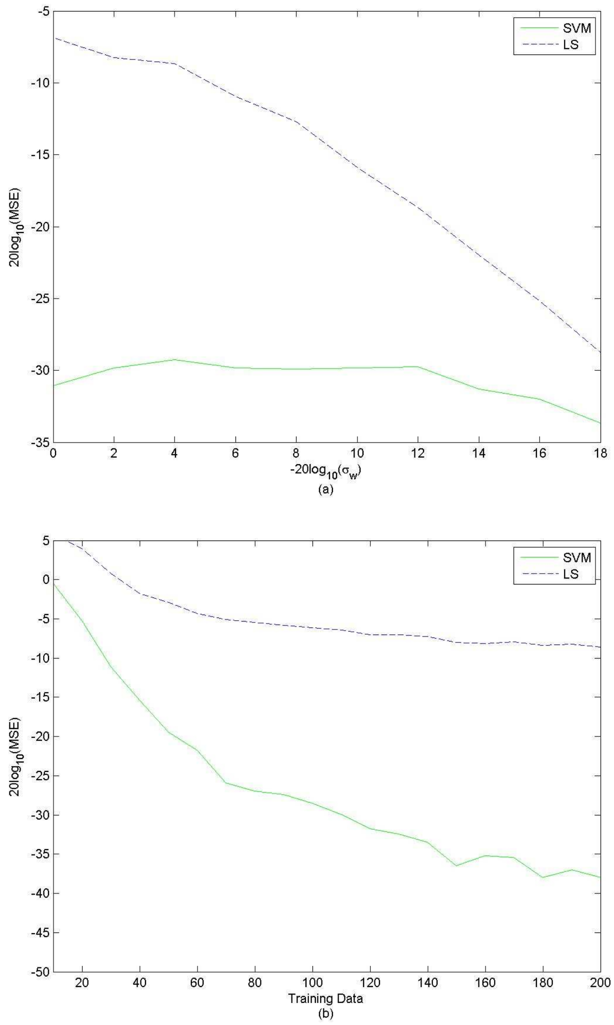


Fig. 5. (a) The MSE as a function of additive noise of power $\sigma_w$ for SVR-ARMA$_{2k}$ model. (b) The MSE as a function of training data for SVR-ARMA$_{2k}$ model, where $\sigma_w = 1$.

TABLE I
THE nMSE OF BESSEL EQUATION.

| | | LS method |
|---|---|---|
| SVR | -0.927 | -0.927 |
| SVM-ARMA$_{2k}$ | -0.935 | -0.935 |
| SVR-ARMA $_{2k}$ | -1.045 | -1.045 |
| SVM-ARMA $_{4k}$ | -1.246 | -1.045 |
| SVR-ARMA $_{4k}$ | -1.262 | -1.045 |

TABLE II
THE nMSE OF BESSEL EQUATION WITH GAUSSIAN NOISE.

| | | LS method |
|---|---|---|
| SVR | 0.003 | 0.003 |
| SVM-ARMA$_{2k}$ | 0.003 | 0.003 |
| SVR-ARMA $_{2k}$ | 0.003 | 0.003 |
| SVM-ARMA $_{4k}$ | 0.003 | 0.003 |
| SVR-ARMA $_{4k}$ | 0.003 | 0.003 |

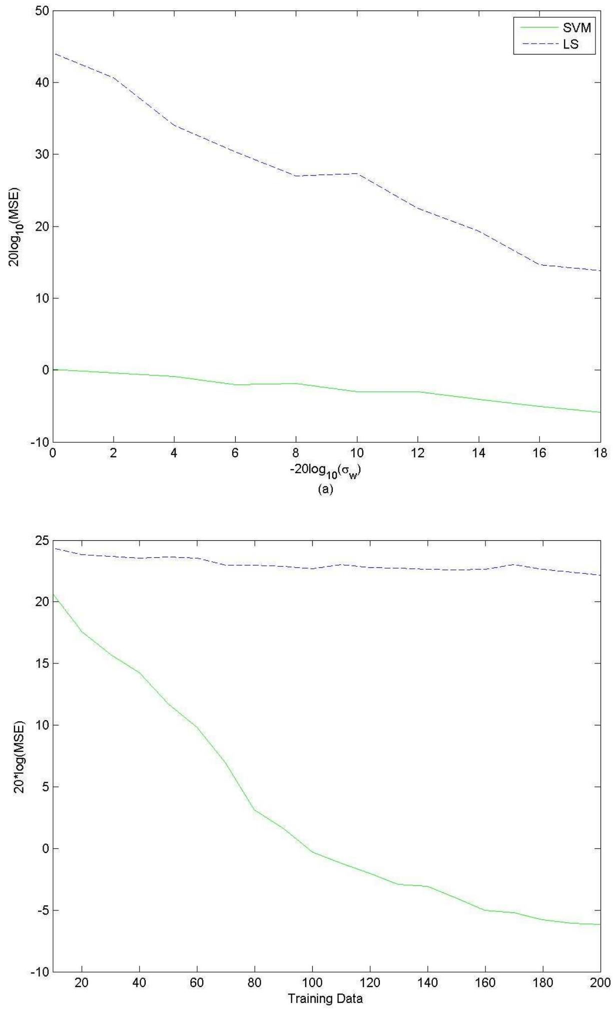


Fig. 6. (a) The MSE as a function of additive noise of power $\sigma_w$ for SVM-ARMA$_{4k}$ model. (b) The MSE as a function of training data for SVM-ARMA$_{4k}$ model, where $\sigma_w = 1$.

Therefore, we may conclude that SVM-ARMA methods provide as good results for Bessel difference equation as the best method LS, with and without additive Gaussian noise.

## V. CONCLUSION

This paper has presented a full family of SVM-ARMA methods for nonlinear system identification in RKHS. These methods are proposed by taking the advantage of composite kernel, in which dedicated mappings are used for input, output and cross terms. Simulation results show the performance of the different SVM-ARMA models and compare it with the least square method.

## REFERENCES

[1] L. Ljung, *System Identification. Theory for the User*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[2] J. Antari,R. Iqdour, S. Safi, A. Zeroual,A. Lyhyaoui, *Identification of Quadratic Non Linear Systems Using Higher Order Statistics and Fuzzy Models*, IEEE International Conference on Acoustics, Speech and Signal Processing. Proceedings of ICASSP 2006.

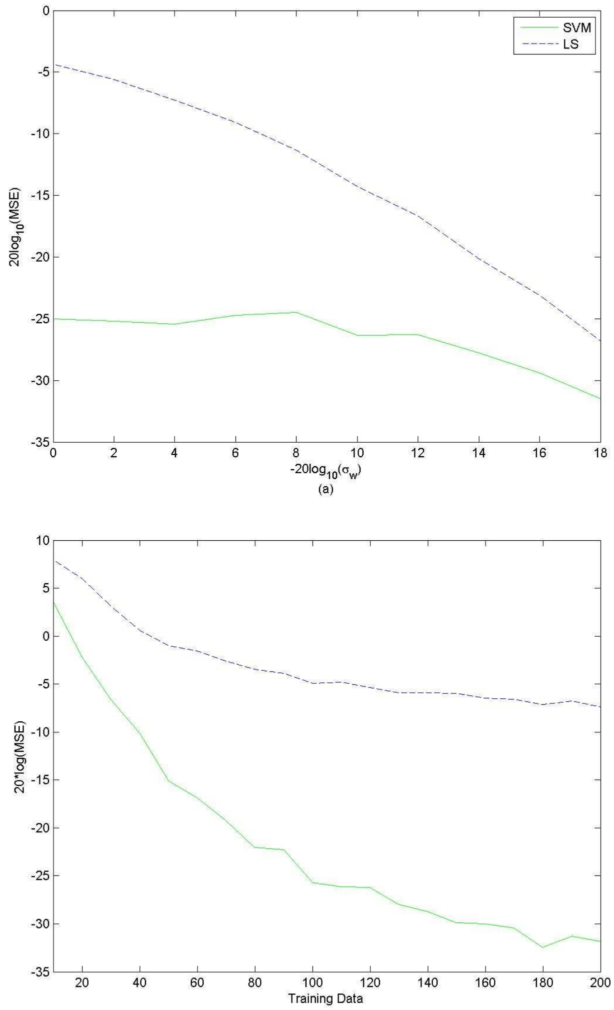


Fig. 7. (a) The MSE as a function of additive noise of power $\sigma_w$ for SVR-ARMA$_{4k}$ model. (b) The MSE as a function of training data for SVR-ARMA$_{4k}$ model, where $\sigma_w = 1$.

[3] M. Ibnkahla, *Nonlinear System Identification Using Neural Networks Trained with Natural Gradient Descent*, EURASIP Journal on Applied Signal Processing 2003, pp.1229-1237.

[4] M. Kaneyoshi,H. Tanaka, M. Kamei,H. Farata, *New System Identification Technique Using Fuzzy Regression Analysis*, International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 1990, pp. 528-533.

[5] A.J. Smola,B. Schölkopf, *A Tutorial on Support Vector Regression*, ESPRIT, Neural computational theory. NeuroCOLT2 NC2-TR-1998-030, 1998.

[6] N.Cristianini, J.Shawe-taylor, *An Introduction to Support Vector Machines, and Other Kernel-Based Learning Methods*, Cambridge press, 2000.

[7] G. Camps-Valls, J.L. Rojo-Álvarez, M. MartínezRamón, *Kernel Methods in Bioengineering, Signal and Image Processing*, Hershey, PA: Idea Group Inc., 2006.

[8] K.R.Müller and al., *Predicting Time Series with Support Vector Machines. Articial Neural Networks*, ICANN'97, pages 999 - 1004, Berlin, 1997.

[9] J.L. Rojo-Álvarez, M. Martínez-Ramón, A.R. Figueiras-Vidal, M. de-Prado Cumplido, A. Artés-Rodriguez, *Support Vector Method for Robust ARMA System Identification*, IEEE Trans. Signal Process., vol. 52, no. 1, pp. 155-64, Jan. 2004.

[10] ¸M. Martínez-Ramón,J.L. Rojo-Álvarez,G. Camps-Valls, J. Muñoz-Marí, Á. Navia-Vázquez, E. Soria-Olivas,A.R. Figueiras-Vidal, *Support Vector Machines for Nonlinear Kernel ARMA System Identification*, IEEE Tans. Neural Networks, vol. 17, no. 6, pp 1617-1622, Nov. 2006.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# The Expert System Approach in Development of Loosely Coupled Software with Use of Domain Specific Language

Piotr Grobelny
University of Zielona Gora,
Faculty of Electrical Engineering, Computer Science and Telecommunication,
Podgorna 50, 65-246 Zielona Gora, Poland
Email: P.Grobelny@weit.uz.zgora.pl

*Abstract*—**This paper addresses the problem of supporting the software development process through the artificial intelligence. The expert systems could advise the domain engineer in programming without the detailed experience in programming languages. He will use and integrate, with the help of deductive database and domain knowledge, the previously developed software components to new complex functionalities. The Service Oriented Architecture (SOA) and loosely coupled software allow to fulfill these requirements. The objective of this document is to provide the knowledge representation of atomic Web Services which will be registered as the facts in the deductive database as well as the inferring techniques. Also, the use of Domain Specific Language (DSL) for modeling domain engineer's requests to the expert system will be considered within this document.**

## I. Introduction

THE aim of this document is to propose a new approach of software development supported by the artificial intelligence. The loosely coupled software, especially the Web Services go towards the need of developing software families through domain engineer which has no detailed experience in computer programming, but has strong expert knowledge. This process could be supported by expert systems.

The background of the consideration is the domain engineering approach [7] which relies on developing software families from reusable components which are parts of common domain system. In the future, the software can be named service-ware, where all resources are services in a Service Oriented Architecture (SOA). The main idea of this approach is that business processes engineer operates on atomic services, not on the software or hardware that implements the service [8].

The method proposed within this paper could be used in large companies enabled on SOA for realizing business processes management (BPM) applications. Web Services are considered a promising technology for Business-to-Business (B2B) integration. A set of services from different providers can be composed together to provide new complex functionalities.

### A. Concept

The expert system plays the role of decision supporting system. Its task is to provide the proposition of complex service (workflow of atomic Web Services) basing on the domain engineer's request explained by means of Domain Specific Language (DSL). The facts in the deductive database are delivered by software developer who implements new functionalities fashioned as the Web Services compliant with enterprise SOA infrastructure. Software developer registers the atomic service model into facts database and also the service instance in SOA registrars (see concept overview in [1]).

The author of this paper proposed in previous work [2] the proof of concept prototype based on the Java framework for intelligent discovery and matchmaking atomic Web Services within integrated workflow called complex service. Thus, the problem of knowledge representation in Services Oriented Architecture as well as usage of DSL will be considered in next sections.

### B. Problem Statement and Challenges

The issue of writing computer program through other computer program is very idealistic challenge, so it seems to be realistic when some assumptions have been fulfilled. The loosely coupled software based on a collection of Web Services that communicate with one another within the distributed systems, which are self-contained and do not depend on the context or state of the other services, allows for discovery of new program functionalities by expert system. The next assumption is that all actors described in concept should use common domain name space (domain objects) expressed through domain ontologies (for instance Web Service Modeling Ontology [9]).

The aim of research work described within this document is to provide the sufficient knowledge representation system which consists of rules, facts and Domain Specific Language. The properly defined service models, which involve interface description, semantic specification as well as information about service quality (QoS) and non-functional properties, registered as the facts in expert system will enable inferring knowledge about enterprise software resources by domain engineer and matchmaking them as new applications.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## II. Related work

The author of [10] describes the semantic service specification, which is the basis for the composition of services to application service processes. Semantic-specified services are the condition for the development of complex functionality within application service processes. The first requirement of the semantic service specification is an existing domain ontology, which describes the domain specific concepts as well as associations and attributes of these concepts. A further requirement for the description of the semantic service specification is a unified description language. The F-Logic language [11] and its extension called Flora-2 [12] have been used. F-Logic is a deductive, object oriented database language which combines the declarative semantics and expressiveness of deductive database languages with the rich data modeling capabilities supported by the object oriented data model [10].

The author of this paper proposes other approach to explain the service models using Java language expressions. The main objective for this solution is to combine in one programming language: knowledge about services, expert system (rule engine) compliant with JSR-94 specification (implementation of the Java Rule Engine API, which allows for support of multiple rule engines [13]) as well as J2EE middleware which is the powerful development platform for Services Oriented Architecture [15]. In the previous paper author proposed the architecture for complex services prototyping and proven the feasibility of this approach on the Java platform using the developed prototype [2].

Domain Specific Language is the way of extending the rule to problem domain. In addition, DSL can be used to create front-ends to existing systems or to express complicated data structures. A DSL is a programming language tailored especially to an application domain: rather than being for a general purpose, it captures precisely the domain's semantics [22]. DSL can act as "patterns" of conditions or actions that are used in rules, only with parameters changing each time [19]. Rules expressed in Domain Specific Language have human-readable form and match the expression used by domain experts [22]. The domain engineer models the request to the deductive database as one rule instead of a lot of source code lines and nested loops in structural programming languages or SQL statements.

## III. Implementation

Web Services are software applications with public interfaces described in XML. A proper service description answers three questions about a service: what the service does (including its non-functional description), where it is located, and how it should be executed [16]. The Fig. 1 presents the simplified atomic service model proposed by author of this paper which answers these questions.

All model elements has been described in details in previous author's paper [1] therefore only some Quality of Service (QoS) generic parameters such as execution price and execution duration  will be considered to illustrate the problem of making decisions in fuzzy surroundings as well as usage of Domain Specific Language.

This paper presents the object-oriented approach to usage rule- and model based expert system [23][6] in development of loosely coupled software on Java platform. The author delivers the solution for creating the object-oriented knowledge databases (deductive databases) which consist of atomic service models stored as the facts as well as for inferring the knowledge through rules.

### A. Atomic Service Models as Facts in Knowledge Database

All pieces of information stored in deductive database establish the knowledge representation system. In previous work [1][2] the author proposed the decision tables formalisms to explain the facts as the production rules. Decision tables specify what decisions should be made when some conditions are fulfilled [4].

The knowledge representation system K which distinguishes the condition and decision attributes can be called a decision table:

$$K=(U,A,C,D) \qquad (1)$$

Where U is a nonempty, finite set called universe, A is a nonempty set of primitive attributes and C, D ⊂ A are two subsets of attributes called condition and decision attributes. Any implication

$$\phi \rightarrow \psi \qquad (2)$$

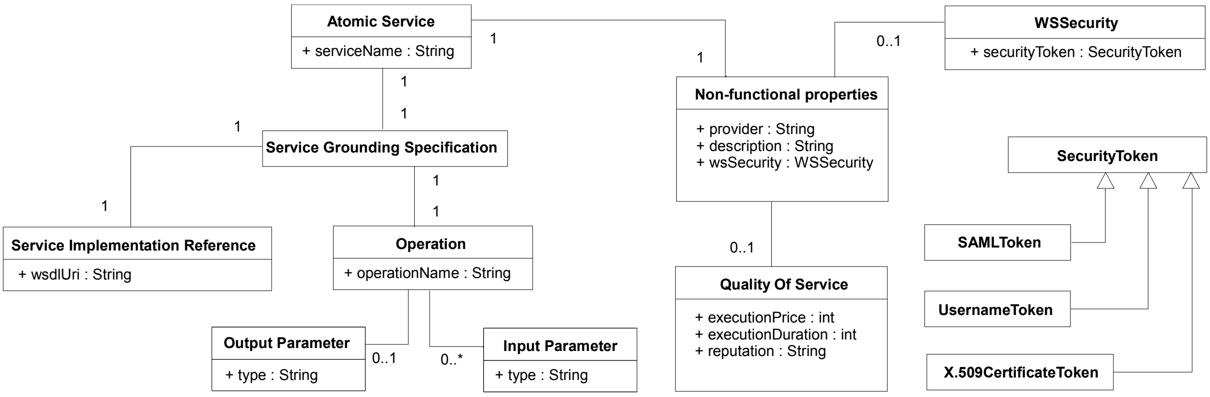


Fig 1. The simplified model of atomic service

is considered as the decision rule and Φ, Ψ are called predecessor and successor respectively.

If $\Phi \rightarrow \Psi$ is decision rule and P contains all attributes occurring in Φ (condition attributes) and Q contains all attributes occurring in Ψ (decision attributes) then this decision rule can be called PQ-rule.

Let's consider the real decision table (see Table I) which represents the knowledge system from geographic information systems (GIS) domain in Services Oriented Architecture and the facts are explained as the PQ-rules. The use case scenario and the services landscape were described within author's previous paper [2].

The columns P1-P6 represent the condition attributes and column Q1 represents the decision attribute of the PQ-rule. These PQ-rules are stored as facts in expert system database. The formula (3) formalizes a possible representation of PQ-rule from Table I in accordance to the formula (2).

$$P1=getMap \ \wedge \ P2=\{Coordinates\} \ \wedge \ P3=Map \ \rightarrow \ Q1=GisMap \quad (3)$$

The author of this paper prepared the facts database in terms of PQ-rules regarding formula (3) and Table I as the Java class which is loaded into the working memory of expert system. The previous document [1] presents the implementation of the selected fact which expresses the atomic service model with compliance to Jboss Drools [19] object-oriented expert system.

### B. Making Decisions in Fuzzy Surroundings

Respectively to formula (3) the facts regarding the membership functions could be explained as the PQ-rules as shown in Table II. Let's consider the example membership functions of fuzzy sets "low execution duration" and "average execution price". These functions should be provided by experts and can be explained in object-oriented manner as suggested in [3].

In the paper [1] has been presented the Java object with implemented membership function for QoS parameter "execution duration" and DSL pattern "low".

The fuzzy sets theory [5][14] allows for making decisions in fuzzy surroundings. The fuzzy set A in certain nonempty space X is defined as collection of couples as shown in (4).

$$A=\{(x,\mu_A(x));x\in X\} \quad (4)$$

where

$$\mu_A : X \rightarrow [0,1] \quad (5)$$

is the membership function of fuzzy set A.

Formula (6) represents the **symbolic** presentation of fuzzy set A for the finite number of elements X = {$x_1$, ... , $x_n$}, A⊆ X

$$A=\frac{(\mu_A(x_1))}{x_1}+\frac{(\mu_A(x_2))}{x_2}+\ldots+\frac{(\mu_A(x_n))}{x_n}=\prod_{i=1}^{n}(\frac{(\mu_A(x_i))}{x_i})$$ ( 6

)

where

$$\frac{(\mu_A(x_i))}{x_i}, i=1,\ldots,n \quad ( 7 )$$

denotes the pair ( 8 ) and symbol "+" has the symbolic character, as well.

$$(x_i,\mu_A(x_i)), i=1,\ldots,n \quad ( 8 )$$

The fuzzy surroundings consists of fuzzy constraints C and fuzzy decision D. Let's consider the set of options $X_{op}$={x}. The fuzzy constraint is defined as fuzzy set C specified on $X_{op}$ and described through membership function $\mu_c$: $X_{op} \rightarrow$ [0,1].

Let's consider the *n* fuzzy constraints $C_1$ , … $C_n$ where *n* >1. The fuzzy decision D is determined though formula (9).

$$D=C_1\wedge C_2\wedge\ldots\wedge C_n \quad ( 9 )$$

taking into account ( 10 )

$$\mu_D(x)=T\{\mu_{C_1}(x),\ldots,\mu_{C_n}(x)\}, x\in X_{op} \quad ( 10 )$$

where T denotes the t-norm MIN operator as shown in (11)

$$\mu_D(x)=MIN(\mu_{C_1}(x),\ldots,\mu_{C_n}(x)) \quad ( 11 )$$

The maximal decision is the option $\dot{x}$ ∈ X $_{op}$ selected in the manner described in formula ( 12 )

$$\mu_D(\dot{x})=max(\mu_D(x)), x\in X_{op} \quad ( 12 )$$

The author took under consideration a situation, when the inference engine proposed three services from Table I regarding to defined constraints $C_1$ - "low execution duration" and $C_2$ - "average execution price". These services establish the options collection $X_{op}$={"MapProvider", "PrintMap", "GisMap"} which are denoted respectively $x_1$, $x_2$, $x_3$. The task of the reasoning process is to take the maximal decision

TABLE I.
THE EXAMPLE OF DECISION TABLE WITH SELECTED ATOMIC SERVICE MODEL ATTRIBUTES

| **Operation** | **Input Parameters** | **Output Parameter** | **Provider** | **Execution Price** | **Execution Duration** | **Service Name** |
|---|---|---|---|---|---|---|
| *P1* | *P2* | *P3* | *P4* | *P5* | *P6* | *Q1* |
| getMap | {Coordinates} | Map | TeleAtlas | 7$ | 24ms | GisMap |
| printMap | {Coordinates} | Map | GIS Atlas | 10$ | 12ms | PrintMap |
| provideMap | {Location} | Map | GIS Maps | 110$ | 49ms | MapProvider |
| drawSegment | {Coordinates, Coordinates, Map} | Segment | GIS Company | 0$ | 5ms | DrawSegment |
| computeDistance | {Coordinates, Coordinates} | Distance | ITS | 0$ | 1ms | ComputeSegmentDistance |

TABLE II .
THE EXAMPLE OF DECISION TABLE WITH MEMBERSHIP FUNCTIONS

| QoS | DSL Pattern | Membership Function Object |
|---|---|---|
| *P1* | *P2* | *Q1* |
| execution duration | low | LowExcutionDurationMembership Func |
| execution price | average | AverageExcutionPriceMembership Func |

$\dot{x}$ as the best proposition regarding the constraints $C_1$ and $C_2$.

The formulas ( 13 )( 14 ) specify the fuzzy constraints $C_1$ and $C_2$ using symbolic notation from ( 6 ) which points out the computed certainty factor (CF) for each option.

$$C_1 = \frac{0{,}024}{x_1} + \frac{0{,}95}{x_2} + \frac{0{,}649}{x_3} \qquad (13)$$

$$C_2 = \frac{0{,}799}{x_1} + \frac{1{,}0}{x_2} + \frac{0{,}4}{x_3} \qquad (14)$$

Regarding the formulas ( 9 ) and ( 10 ) the fuzzy decision D could be denoted as shown in formula ( 15 ).

$$D = \frac{0{,}024}{x_1} \wedge \frac{0{,}95}{x_2} \wedge \frac{0{,}4}{x_3} \qquad (15)$$

The maximal decision $\dot{x}$ assigned basing on formula (12) points the option $x_2$—“PrintMap” as the best proposition of atomic Web Service in context of defined constraints.

### C. The Use of Domain Specific Language

The domain engineer models the request to the deductive database as the production rules presented in formula (2) manner. These rules allow the experts to express the logic in their own terms [19] of Domain Specific Language (DSL):

```
rule "Service options"
when
    There is a service where
        - output parameter equals "Map"
    There is a service with "low" execution dura-
    tion
then
    Create service options set
end

rule "Make decision in fuzzy surroundings"
when
    There is maximal decision for service options
    set
then
    Propose the best service
end
```

The DSL expressions should be transformed to the notation accepted by shell of expert system:

```
[condition]
There is a service where=
$as: AtomicService ($qos : qos, $serviceName :
serviceName)

[condition]
- output parameter equals "{value}"=
outputParameter == "{value}"

[condition]
There is a service with "{value}" execution dura-
tion=
FuzzyRules ($patternED : pattern == "{value}",
$featureED : feature == "execution duration",
$membershipFuncED : membershipFunc,
eval($membershipFuncED.value($qos.executionDura-
tion) > 0));

[consequence]
Create service options set=
ArrayList al = new ArrayList();
al.add(new Double($membershipFuncED.value($qos.ex-
ecutionDuration)));
al.add(new Double($membershipFuncEP.value($qos.ex-
ecutionPrice)));
insert (new Option ($as, al));
```

The result of `Service options` reasoning rule is the option collection X $_{op}$ with values of the certainty factor (CF):

```
– Option: MapProvider
  low execution duration CF: 0.024,
  average execution price CF: 0.799

– Option: PrintMap
  low execution duration CF: 0.95,
  average execution price CF: 1.0

– Option: GisMap
  low execution duration CF: 0.649,
  average execution price CF: 0.4
```

The outcome of the `Make decision in fuzzy surroundings` rule is the best service proposition (`Proposed service: PrintMap, max membership level: 0.95`) in context of defined constraints established basing on the maximal decision formula as shown in (15).

As the expert system the JBoss Drools [9] rule engine based on the Rete algorithm [20] has been used. Drools implements and extends the Rete algorithm which is called ReteOO, what signifies that Drools has an enhanced and optimized implementation of the Rete algorithm for object-oriented systems [21].

## IV. CONCLUSION

The presented approach allows to support the domain engineer in developing applications from business processes management area. The domain engineer has no detailed experience in computer programming, but has strong expert knowledge. He can model the requests to the deductive database as the production rules in human-readable format with usage of Domain Specific Languages instead of several lines and nested loops of programming language code. The author discussed within this paper the knowledge representation in SOA and making decisions in fuzzy surroundings with the help of DSL patterns.

The further research will be focused on refinement of reasoning process with usage of other techniques of the artificial intelligence, development of domain specific languages for

GIS domain as well as discovery and matchmaking workflow of complex services.

The use of expert system seems to be a promising way of programing loosely coupled software systems by the domain experts. Also the Java platform strongly focused on Web Services as well as available object-oriented deductive databases allow for application of proposed solutions in business processes management.

## References

[1] P. Grobelny, "Knowledge representation in services oriented architecture", in *Przeglad Telekomunikacyjny,* 6/2008, SIGMA NOT, pp. 793-796.

[2] P. Grobelny, "Rapid prototyping of complex services in SOA architecture" in *Conference Archives PTETiS*, vol. 23(1), Warszawa, 2007, pp. 71-76.

[3] D. H. Besset, *Object-Oriented Implementation of Numerical Methods, An Introduction with Java and Smalltalk*, Morgan Kaufamann Publishers, Academic Press, 2001.

[4] Z. Pawlak, *ROUGH SETS Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, 1991

[5] L. Rutkowski, *Metody i techniki sztucznej inteligencji*, Wydawnictwo Naukowe PWN, 2006

[6] A. Niederliński, *Regułowo-modelowe systemy ekspertowe rmse*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 2006

[7] K. Czarnecki, U. Eisenecker, *Generative Programming – Methods, Tools and Applications*, Boston, MA: Addison Wesley, 2000

[8] A. Ekelhart, S. Fenz, A Min Tjoa, E. Weippl, "Security issues for the use of semantic web in e-commerce", in *Business Information Systems, 10th International Conference BIS 2007 proceedings*, W. Abramowicz, Ed., Springer, 2007, pp.1-13.

[9] R. Dumitru, U. Keller, H. Lausen, J. De Bruijn, L. Ruben, M. Stollberg, A. Polleres, C. Feier, C. Bussler, D. Fensel, "Web service modeling ontology" in *Applied Ontology*, 1(1), IOS Press, 2005, pp.77-106.

[10] S. Donath, "Automatic creation of service specifications" in *Net.ObjectDays Proceedings,* 6th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for Networked World, 2005, pp.79-89.

[11] M. Kifer, G. Lausen, Wu J, "Logical foundations of object-oriented and frame-based languages" in *Journal of the Association for Computing Machinery,* 1995.

[12] G. Yang, M. Kifer, C. Zhao "FLORA-2: A rule-based knowledge representation and inference infrastructure for the semantic web", Second International Conference on Ontologies, Databases and Applications of Semantics (ODBASE), Italy, 2003.

[13] A. Toussaint, *Java Rule Engine API™ JSR-94*, Java Community Process, BEA Systems, 2003.

[14] J. Kacprzyk, *Wieloetapowe sterowanie rozmyte*, Warszawa: WNT, 2001.

[15] M. Hansen, *SOA Using Java Web Services,* Person Education Inc., Prentice Hall, 2007.

[16] M. Kowalkiewicz, "Current challenges in non-functional service description – state of the art and discussion on research results", *Net.ObjectDays Proceedings,* 2005, pp. 91-96.

[17] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, *Web Services Description Language (WSDL) 1.1,* World Wide Web Consortium W3C, 2001.

[18] *UDDI Specifications,* http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm , 2007.

[19] M. Proctor, M. Neale, M. Frandsen, S. Griffith, E. Tirelli, F. Meyer, K. Verlaenen, *Drools Documentation*, http://jboss.com, 2008.

[20] C. Forgy, "RETE: A fast algorithm for the many pattern many object pattern match problem" in *Artificial Intelligence,* 19(1), 1982, pp.17-37.

[21] R. Doorenbos, "Production matching for large learning systems (Rete/UL)", Ph.D. thesis, Carnegie Mellon University, 1995.

[22] D. Spinellis, "Notable design patterns for domain-specific languages" in *The Journal of Systems and Software,* 56, Elsevier, 2001, pp. 91-99.

[23] A. Niederliński, "An expert system shell for uncertain rule- and model based reasoning" in *Methods of Artificial Intelligence,* Gliwice, 2001.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Coalition formation in multi-agent systems—an evolutionary approach

Wojciech Gruszczyk
Wrocław University of Technology
Computer Engineering
Email: 141044@student.pwr.wroc.pl

Halina Kwaśnicka
Wrocław University of Technology
Computer Engineering
Email: halina.kwasnicka@pwr.wroc.pl

***Abstract*—The paper introduces solution of Coalition Formation Problem (CFP) in Multi-Agents Systems (MAS) based on evolutionary algorithm. The main aim of our study is to develop an evolutionary based algorithm for creation of coalitions of agent for solving assumed tasks. We describe the coding schema and genetic operators such as mutation, crossover and selection that occurred to be efficient in solution of CFP. Last part of the document provides a brief comment on our research results.**

***Index Terms*—Coalition formation, evolutionary algorithm, multi–agent systems**

## I. Introduction

CONCEPT of agents (term agent will be used without distinction between software and hardware agents) is strongly connected with artificial intelligence (AI). Because the term “agent” is defined in literature in many different ways we will use definition first proposed in [7]:

**Agent** is software or hardware computer system that is/has:

1) Autonomous: agent takes actions without interference of a human and has control over taken actions,
2) Social ability: agents communicate (between themselves and/or with people),
3) Reactivity: agents have some perception of environment that they are part of and may react to changes in the environment,
4) Activity: agents may take actions to change their environment in order to achieve their goals.

When the environment contains at least two agents we talk about multi-agent system (MAS). Very often MAS are distributed. Many aspects of such systems have been widely discussed in literature. In this paper we present a new solution of the problem of coalition formation (CFP).

### A. Coalition Formation

Many tasks cannot be completed by a single agent because of limited resources or capabilities of agents. Very often, even if a task may be completed by a single agent, his performance may occur too low to be acceptable. In such a situation agents may form groups to solve the problem by cooperation. Many organizational paradigms have been distinguished in the context of MAS (see [9]). This work is focused on coalitions—groups of cooperative agents, working together on a given task, short-lived and goal-directed, being a flat structure. Initially agents are independent and do not cooperate. When they cannot complete their tasks individually agents may exchange information and try to form coalitions which gives them best efficiency (of course the efficiency must be defined in terms of solved problem).

Evaluation of all possible shapes of coalitions depends exponentially on the number of agents. For example having $m$ agents and $n$ tasks to solve all $n^m$ coalitions (with each coalition delegated to a particular task) must be evaluated to guarantee that the best coalition shape was found. Finding the optimal partition of agents set by checking the whole space may occur too expensive (in terms of time). Short lifetime of coalition may lead to a situation when time of computation is far longer than the time of existence of a particular coalition. Therefore it may create a bottleneck of a MAS. Many methods have been proposed to solve CFP. Further part of the introductory chapter will give a short summary of them.

### B. CFP—a short overview

*1) Any-time solution with worst case guarantee:* Original work given by [8] presents any-time solution with worst case guarantee. The idea is based on a remark, that searching subset of possible coalition shapes that contains all possible subsets of the set of agents may guarantee quality of the solution. That is why the best solution must consist of already searched and evaluated subsets (best solution is not worse than the best solution found so far). To make the idea clear [8] suggested representing all partitions as a graph of coalition structures (Fig. 1):

Level $LV_i$ denotes that the structures belonging to this level consist of i coalitions. Searching $LV_1$ and $LV_2$ assures that all subsets have been checked and may provide guarantee on the result. Then [8] suggests searching bottom up (in the picture levels $LV_4$, $LV_3$). Checking all coalition structures leads to brute force search and worst case guarantees are low before huge amount of the space has been searched (as shown in [5]).

*2) Distributed algorithm:* Distributed algorithms for solving CFP are proposed (among others) in [2] and [5]. The solution proposed in [5] (compared to other distributed methods) significantly minimizes efforts on communication between agents.

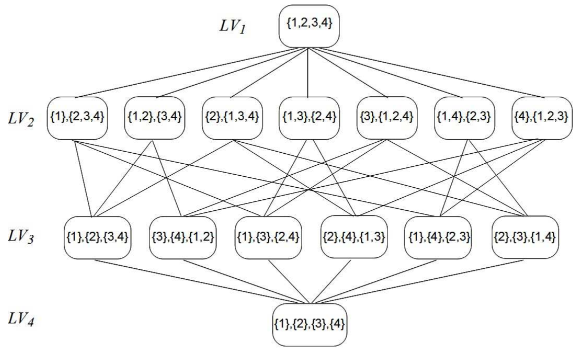


Fig. 1. Coalition structure graph for four agents

*3) Genetic algorithm based method:* In [1] the authors presents a genetic algorithm where two-dimensional, binary chromosome coding is used. The method uses so called *two dimensional "or" crossover operator* that seems to be troublesome, because after using the operator we have to review child chromosome in order to "repair" it when the operator breaks the rules of representation assumed in the method. Inspired by [1] we decided to propose an alternative chromosome coding and operators to solve CFP.

Good introduction to MAS provides [3]. Brief descriptions of organizational paradigms other than coalition of agents are presented in [9]. Environments of self-oriented, conflicting agents are described in [6]. Comprehensive introduction to evolutionary programming is provided in [4].

The main aim of our study is to develop a method of coalitions formation in multi-agent systems using evolutionary approach. The method should combine good efficiency in solving CFP with natural chromosome coding and genetic operators.The paper is structured as follows. Next section will introduce formal representation of agents and coalitions. Third chapter will cover implementation issue and experimental results. The last part of the paper concludes the results and point further research directions.

## II. The proposed method

In this section we present a formal model of our problem and the proposed evolutionary algorithm.

### A. Model

At the beginning we assumed as follows:

1) Agents solve a finite number of tasks,
2) Each agent has some ability (resources) to solve each task,
3) Each task requires some abilities (resources) to be solved,
4) Particular agent may have insufficient abilities (resources) to solve a task,
5) Agents are cooperative,
6) Agents are altruistic—their own good is less important than the good of the system as a whole,
7) Each agent must contribute to exactly one coalition,
8) We want to solve as many tasks as possible with the given set of agents.

According to these assumptions the following model of the problem has been proposed:

Set of tasks is represented as a vector T:

$$T = < T_1, \ldots, T_n >, T_i \in (\mathbb{N}_+ \setminus \{1\}), i > 1$$

Value $t$ at position $T_i$ means that to solve task $i$ $t$ units of resource (ability) are needed. We assume that $i > 1$ because for $i = 1$ the problem is trivial.

Let $A$ be a set of agents:

$$A = \{A_1, \ldots, A_k\}$$

where each agent $A_i$ is represented as:

$$A_i = < U_1, \ldots, U_n >, U_i \in \mathbb{N}, i > 1$$

Value $p$ at position $U_j$ means that agent $i$ has abilities (resources) of $p$ to solve task $j$.

Set $K$ representing partition of set $A$ with properties given below will be the result of proposed algorithm:

- Each element $C_i \in K$ will be a coalition,
- Each element $C_i$ will be a set of agents such that $C_i \subseteq A$ (in particular $C_i$ may be empty),
- For each $C_i, C_k \in K$: $C_i \cap C_k = \emptyset, i \neq k$
- $\bigcup_{i=1}^{n} C_i = A, n = card(K)$
- Membership of an agent in a coalition $i, 1 \leq i \leq n$ means that the agent contributes to solution of task $i$.
- $K$ represents partition that achieved the best (the highest) value of fitness function (described further).

We defined total ability of a coalition $i$ to solve its task as a simple sum of abilities (resources) of its (coalition's) members' abilities to solve the task. Formally total ability $\delta_i$ is given as:

$$\delta_i = \sum_{A_t \in C_i} U_i[A_t]$$

We say that task $i, 1 \leq i \leq n$ is solved in a given partition when (for given $i$) $\delta_i \geq T_i$.

Evolutionary algorithm operates on a set of individuals—$V$—each of them representing some partition of set $A$. Each individual is represented as a vector $v_i$:

$$v_i = < d_1, \ldots, d_q >, \text{ where:}$$
$$d_i \in \mathbb{N} \wedge d_i \in [1, card(T)], q = card(A)$$

All genetic operations are conducted on elements (individuals) from set $V$. Both operators used in our method (crossover and mutation) are presented in next subsection.

### B. Evolutionary algorithm

Evolutionary algorithm that we developed uses $(\mu + \lambda)$ strategy (for detailed information about evolutionary strategies see [4]). We use $\mu = \lambda$. To preserve the best (so far) solution individual with the highest value of fitness function always survives (is added to child population).

Figure 2 presents block diagram presenting main loop of the algorithm.

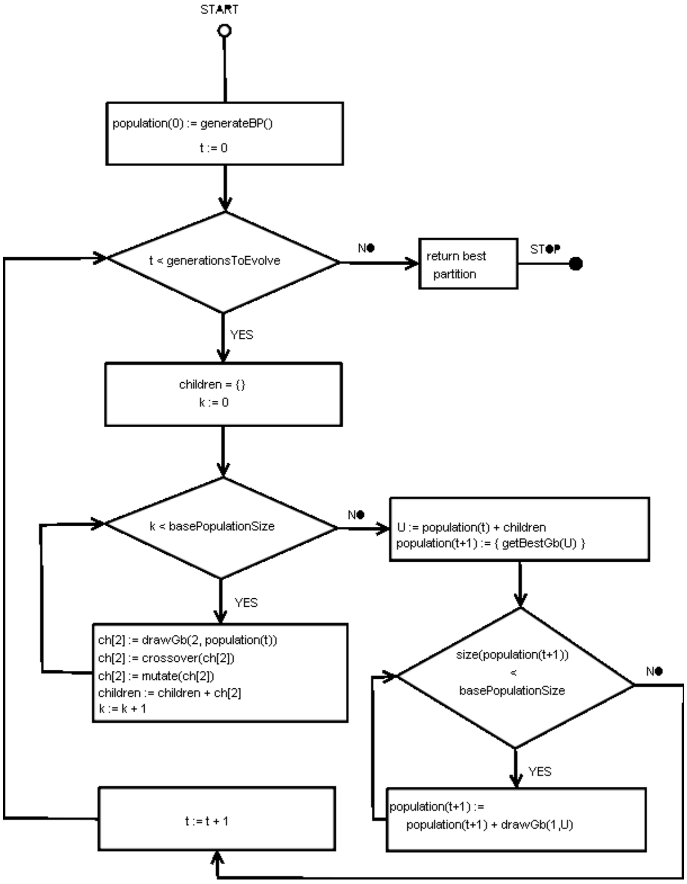


Fig. 2. Evolutionary algorithm block diagram

A few elements on the diagram (Fig. 2) need comment:

- Function drawGb(2, population(t)) draws two individuals from population(t) with returning,
- Function getBestGb(U) gets individual with the highest fitness function value, suffix Gb means that this operation is conducted with returning.

### *C. Operators, fitness function, base population*

*1) Crossover:* We use standard two-point crossover operator. Chromosomes to cross are drawn (chosen stochastically) with giving back with probability proportional to their fitness (fitness and its function are discussed later). Both produced chromosomes are added to child population. Proposed crossover operator preserves the property that each agent (in a child chromosome) belongs to exactly one coalition. Therefore we do not need to repair the chromosome. Two-point crossover (and many more) are discussed in [4].

*2) Mutation:* Mutation is carried out with a given probability $p$ on a single gene. The operator randomly chooses whether to mutate the gene. Mutation randomly changes the coalition to which the agent represented by the gene is assigned. Alike crossover, mutation does not break the chromosome therefore no repairs are needed.

*3) The initial population:* Initial population (the first generation) is generated randomly. Its size is given arbitrarily and all created chromosomes are correct in terms of definitions given in previous chapters.

*4) Fitness function:* A fitness function is essential for the method. As we want to maximize the number of tasks solved by coalitions of agents following assumptions about the function have been made:

- An individual should be punished for non-completion of a task,
- And individual should be punished for exceeding the required abilities (resources) by a coalition ($\delta_i > T_i$),
- Punishment for non-completion of a task should be more harmful than the punishment for exceeding the boundary,
- Fitness function must be non-negative.

Based on the above remarks we designed the following way of fitness calculation (Algorithm 1):

**Algorithm 1** Calculating fitness function value

```
$fitVal \leftarrow numberOfTasks$
for AllCoalitions : Ci do
    if $\delta_i < T_i$ then
        $fitVal \leftarrow fitVal - 1.2 + max(0.2, \delta_i/T_i)$
    else
        if $T_i \neq 0$ then
            $fitVal \leftarrow fitVal - min(0.05, (\delta_i - T_i)/T_i)$
        else
            $fitVal \leftarrow fitVal - 0.05$
        end if
    end if
end for
return fitVal
```

All coefficients in the above algorithm were chosen by manual testing of various values. Given coefficients occurred to be satisfactory in solved problem but should be treated as a hint not as a rule.

At this point it is worth mentioning that the given fitness function is not perfect. It may happen in some situations that some coalition's structure $A$ solving more tasks than coalition $B$ has lower fitness value. It is caused by the punishment for exceeding task's boundary of required resources (abilities). Such a situation usually does not spoil algorithm's results, nevertheless, it is discussed further together with experimental results.

## III. Implementation, results of experiments

Our implementation language was Java. Neither evolutionary algorithm nor MAS frameworks were used. Usage of Strategy pattern occurred to be useful while trying different operators, population generators, etc. See [10] for comprehensive information about design patterns.

Different types of test data have been used. We prepared:

- 2 types of agent populations (random and predefined),
- 2 sizes of agent population,
- 2 sizes of task set,
- 2 sizes of base population.

Brief comment of test data is provided in the next section. After a series of initial test with different values of mutation probability and number of generations we decided to use:

- Probability of mutation: $1\%$,
- Number of generations: $500$.

Given values provided good results, nevertheless, (like for fitness function) they should be treated as a hint.

*5) Agent populations, tasks requirements:* We used two sizes of agent populations containing respectively: 20 agents (solving 5 tasks) and 30 agents (solving 10 tasks). Each population size was tested in two variants: predefined and generated. Predefined population of 20 agents was defined as:

$$\begin{array}{c}
A_{20} = \{ \\
<2,1,1,1,1>, <2,1,1,1,1>, \\
<2,1,1,1,1>, <2,1,1,1,1>, \\
<1,2,1,1,1>, <1,2,1,1,1>, \\
<1,2,1,1,1>, <1,2,1,1,1>, \\
<1,1,2,1,1>, <1,1,2,1,1>, \\
<1,1,2,1,1>, <1,1,2,1,1>, \\
<1,1,1,2,1>, <1,1,1,2,1>, \\
<1,1,1,2,1>, <1,1,1,2,1>, \\
<1,1,1,1,2>, <1,1,1,1,2>, \\
<1,1,1,1,2>, <1,1,1,1,2> \\
\}
\end{array}$$

Set of 30 agents was defined as:

$$\begin{array}{c}
A_{30} = \{ \\
<2,1,1,1,1,1,1,1,1,1>, <2,1,1,1,1,1,1,1,1,1>, \\
<2,1,1,1,1,1,1,1,1,1>, <2,1,1,1,1,1,1,1,1,1>, \\
<2,1,1,1,1,1,1,1,1,1>, <2,1,1,1,1,1,1,1,1,1>, \\
<2,1,1,1,1,1,1,1,1,1>, <1,2,1,1,1,1,1,1,1,1>, \\
<1,2,1,1,1,1,1,1,1,1>, <1,2,1,1,1,1,1,1,1,1>, \\
<1,2,1,1,1,1,1,1,1,1>, <1,2,1,1,1,1,1,1,1,1>, \\
<1,2,1,1,1,1,1,1,1,1>, <1,2,1,1,1,1,1,1,1,1>, \\
<1,1,4,1,1,1,1,1,1,1>, <1,1,4,1,1,1,1,1,1,1>, \\
<1,1,1,4,1,1,1,1,1,1>, <1,1,1,4,1,1,1,1,1,1>, \\
<1,1,1,1,4,1,1,1,1,1>, <1,1,1,1,4,1,1,1,1,1>, \\
<1,1,1,1,1,4,1,1,1,1>, <1,1,1,1,1,4,1,1,1,1>, \\
<1,1,1,1,1,1,4,1,1,1>, <1,1,1,1,1,1,4,1,1,1>, \\
<1,1,1,1,1,1,1,4,1,1>, <1,1,1,1,1,1,1,4,1,1>, \\
<1,1,1,1,1,1,1,1,4,1>, <1,1,1,1,1,1,1,1,4,1>, \\
<1,1,1,1,1,1,1,1,1,4>, <1,1,1,1,1,1,1,1,1,4> \\
\}
\end{array}$$

$T$ vectors for 20 and 30 predefined agents were given as (respectively):

$$T_{20} = <8,8,8,8,8>$$

$$T_{30} = <14,14,8,8,8,8,8,8,8,8>$$

In the case of generated sets of agents we used a generator that (having predefined $T$ vector) prepared such a set that could solve all tasks without a punishment (in terms of fitness function) for exceeding the threshold of needed resources. Each agent in a generated set has positive value of ability to solve any task so it is not obvious to which coalition it should be assigned. Of course the generator does not guarantee that only one optimal solution exists, but it ensures existence of at least one.

We decided to use test data that ensured existence of a solution that solved all tasks. Resources were limited (in fact limits were very rigid) and search spaces were huge (for 10 tasks and 30 agents the whole space consisted of $10^{30}$ possible coalitions!). The choice was suggested by used method which is worth applying to extremely hard combinatorial problems that cannot be solved in acceptable amount of time by deterministic search of the space (e.g. dynamic programming).

*6) The initial population:* We tried two sizes of the initial population for the evolutionary algorithm: 25 and 50 individuals.

*7) Experimental results:* To find optimal parameters for the algorithm we compared four cases:

- Generated test data, base population size: 25/50, number of tasks: 5, number of agents: 20,
- Generated test data, base population size: 25/50, number of tasks: 10, number of agents: 30,
- Predefined test data, base population size: 25/50, number of tasks: 5, number of agents: 20,
- Predefined test data, base population size: 25/50, number of tasks: 10, number of agents: 30.

Results of our experiments are listed in tables I, II, III and IV:

TABLE I

EXPERIMENT I: GENERATED TEST DATA, BASE POPULATION SIZE: 25/50, NUMBER OF TASKS: 5, NUMBER OF AGENTS: 20

| No | Population size | Tasks solved (best) | Fitness function (best) | In generation |
|---|---|---|---|---|
| 1 | 25 | 5 | 4.95 | 16 |
| 2 | | 5 | 4.95 | 35 |
| 3 | | 5 | 5.00 | 22 |
| 4 | | 5 | 4.95 | 17 |
| 5 | | 5 | 5.00 | 17 |
| 6 | | 5 | 5.00 (4.95)[1] | 146 (61)[2] |
| 7 | | 5 | 5.00 (4.90) | 269 (13) |
| 8 | | 5 | 5.00 (4.90) | 46 (8) |
| 9 | | 5 | 5.00 (4.95) | 14 (4) |
| 10 | | 5 | 5.00 (4.85) | 371 (7) |
| 11 | 50 | 5 | 5.00 | 18 |
| 12 | | 5 | 5.00 (4.85) | 55 (6) |
| 13 | | 5 | 5.00 (4.85) | 127 (12) |
| 14 | | 5 | 5.00 (4.80) | 25 (11) |
| 15 | | 5 | 5.00 (4.85) | 54 (13) |
| 16 | | 5 | 5.00 (4.95) | 24 (10) |
| 17 | | 5 | 5.00 (4.90) | 388 (7) |
| 18 | | 5 | 5.00 | 16 |
| 19 | | 5 | 5.00 | 18 |
| 20 | | 5 | 5.00 (4.80) | 35 (8) |

TABLE II
EXPERIMENT II: GENERATED TEST DATA, BASE POPULATION SIZE: 25/50, NUMBER OF TASKS: 10, NUMBER OF AGENTS: 30

| No | Population size | Tasks solved (best) | Fitness function (best) | In generation |
|---|---|---|---|---|
| 1 | 25 | 10 | 9.85 (9.80) | 272 (207) |
| 2 | | 10 | 9.90 (9.80) | 298 (94) |
| 3 | | 9 | 9.65 | 55 |
| 4 | | 9 | 9.70 (9.15) | 429 (37) |
| 5 | | 10 | 9.95 (9.85) | 481 (113) |
| 6 | | 10 | 10.00 | 49 |
| 7 | | 9 | 9.55 (9.45) | 142 (107) |
| 8 | | 10 | 10.00 (9.90) | 325 (269) |
| 9 | | 10 | 9.80 (9.75) | 447 (245) |
| 10 | | 9 | 9.45 | 167 |
| 11 | 50 | 10 | 9.90 (9.85) | 203 (184) |
| 12 | | 10 | 9.90 | 89 |
| 13 | | 10 | 9.90 | 356 |
| 14 | | 10 | 10.00 (9.95) | 210 (139) |
| 15 | | 10 | 9.95 (9.90) | 382 (132) |
| 16 | | 10 | 9.95 (9.85) | 190 (96) |
| 17 | | 10 | 9.95 (9.65) | 90 (34) |
| 18 | | 9 | 9.65 (9.35) | 350 (196) |
| 19 | | 10 | 9.90 | 313 |
| 20 | | 10 | 9.85 | 129 |

TABLE III
EXPERIMENT III: PREDEFINED TEST DATA, BASE POPULATION SIZE: 25/50, NUMBER OF TASKS: 5, NUMBER OF AGENTS: 20

| No | Population size | Tasks solved (best) | Fitness function (best) | In generation |
|---|---|---|---|---|
| 1 | 25 | 4 | 4.05 (4.00) | 101 (18) |
| 2 | | 5 | 5.00 | 98 |
| 3 | | 5 | 5.00 | 178 |
| 4 | | 4 | 4.00 | 38 |
| 5 | | 5 | 5.00 | 124 |
| 6 | | 5 | 5.00 | 80 |
| 7 | | 5 | 5.00 | 73 |
| 8 | | 5 | 5.00 | 126 |
| 9 | | 5 | 5.00 | 86 |
| 10 | | 5 | 5.00 | 82 |
| 11 | 50 | 5 | 5.00 | 189 |
| 12 | | 5 | 5.00 | 94 |
| 13 | | 4 | 4.00 | 27 |
| 14 | | 5 | 5.00 | 42 |
| 15 | | 5 | 5.00 | 315 |
| 16 | | 5 | 5.00 | 107 |
| 17 | | 5 | 5.00 | 37 |
| 18 | | 5 | 5.00 | 114 |
| 19 | | 5 | 5.00 | 33 |
| 20 | | 5 | 5.00 | 41 |

As we can see after 500 generations of algorithm's work in every test case I-IV we obtained solution no worse than $80\%$ of optimal one. Tests I and III in most cases were solved returning optimal partition. For generated test data we obtained $100\%$ accuracy. Average quality of solution in these cases exceeded $90\%$ Two factors had impact on our results:

1) Search space was relatively small ($5^{20}$ compared to $10^{30}$ makes $2^{20} \cdot 10^{10} \approx 10^{16}$ times smaller search space),
2) In the case of generated data it was not so rigid as predefined so more than one optimal solution might have existed.

In both cases the amount of 500 generations was to big number. For generated data optimal solutions were found in less than 100 generations. For predefined data in most cases 200 generations were enough to obtain comparable results.

In test cases number II and IV we achieved accuracy no worse than $80\%$ of optimal solution. Again generated data occurred to be less rigid and gave better results (exceeding $90\%$ of the best solution). For generated data 250 generations of evolutionary algorithm's work occurred to be enough to provide comparable results. Even after 500 generations our algorithm in most cases did not exceeded 8 solved tasks (only 3 times we achieved 9 tasks solved and it took over 270 generations).

Figures 3 and 4 show algorithm's progress. In both cases we put the number of generation on OX and the value of a given function on OY (3 functions are presented and described on the plots). Both plots show a case where despite the growth of fitness of the best individual the number of tasks solved by (current) best solution is lower (on figure 3 we see this situation between 20-th and 200-th generation; on figure 4 about 50-th generation). As mentioned earlier, this situation is caused by punishment used in fitness function. Despite the fact that such a situation is not frequent it may happen that solution with the highest value of fitness function does not provide partition that solves most tasks (even if such partitions were checked).

## IV. CONCLUSIONS AND FURTHER RESEARCH

Our approach occurred to be efficient in solving CFP. Its power is remarkable especially in huge spaces where brute force search and other regular and deterministic search methods cannot be applied because of limited resources (especially time).

Comparing to [1], implementation of GA presented in this paper uses natural coding of chromosome (simple vector rather than complex 2D structure) which makes it easier to use and preserves classic flow of genetic (evolutionary) algorithm. Simple chromosome coding made it possible to use standard mutation and crossover operators. Both of them create individuals that do not break domain constraints (one agent belongs to exactly one coalition) while *two dimensional "or" crossover operator* proposed in [1] enforced checking of

[1] Value in brackets shows the value of fitness function which provided the same number of solved tasks. The value was achieved in a generation which number is shown in brackets in column "In generation". See [2]. The value is given only when the difference between generations is significant.

[2] Situation symmetric to described in [1].

TABLE IV
EXPERIMENT IV: PREDEFINED TEST DATA, BASE POPULATION SIZE: 25/50, NUMBER OF TASKS: 10, NUMBER OF AGENTS: 30

| No | Population size | Tasks solved (best) | Fitness function (best) | In generation |
|---|---|---|---|---|
| 1 | 25 | 8 | 8.45 | 215 |
| 2 | | 8 | 8.09 (7.95) | 154 (68) |
| 3 | | 9 | 9.00 | 273 |
| 4 | | 8 | 8.09 (7.90) | 260 (154) |
| 5 | | 8 | 8.73 (8.30) | 463 (87) |
| 6 | | 8 | 8.44 (7.94) | 87 (24) |
| 7 | | 8 | 8.51 (7.96) | 252 (76) |
| 8 | | 8 | 8.09 (7.95) | 260 (47) |
| 9 | | 8 | 8.09 (7.95) | 391 (57) |
| 10 | | 8 | 8.09 (7.95) | 140 (37) |
| 11 | 50 | 8 | 8.09 (7.90) | 314 (68) |
| 12 | | 8 | 8.44 (7.95) | 252 (77) |
| 13 | | 8 | 8.09 (7.95 ) | 162 (61) |
| 14 | | 8 | 8.01 (7.85) | 495 (393) |
| 15 | | 8 | 8.44 (7.95) | 383 (44) |
| 16 | | 8 | 8.09 | 405 |
| 17 | | 8 | 8.09 (7.95) | 190 (72) |
| 18 | | 9 | 9.00 | 275 |
| 19 | | 8 | 8.09 (7.90) | 402 (87) |
| 20 | | 9 | 9.00 | 407 |

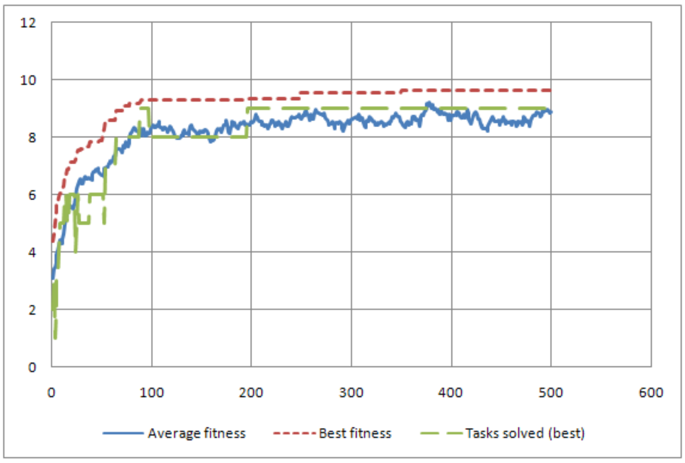


Fig. 3. Algorithm's progress: plot for test II (row number 18)

child chromosomes because they might break the constraint of one-to-one agent to coalition assignment.

Another thing worth mentioning is the initial population. In [1] every coalition structure that does not solve all tasks is assigned fitness of 0. Initial population is generated with revisions whether all tasks are solved and only total efficiency of a whole system (solving all tasks) is being optimized. Such assumption enforces low bounds on available resources (agents may solve all tasks). Our approach does not assume that all tasks must be solved (we do not know whether even a single one can be solved at all). Therefore our initial population is generated randomly, without further revisions. Main optimization target of our method is the number of solved tasks which seems to be very important in real-time, strongly constrained domains.

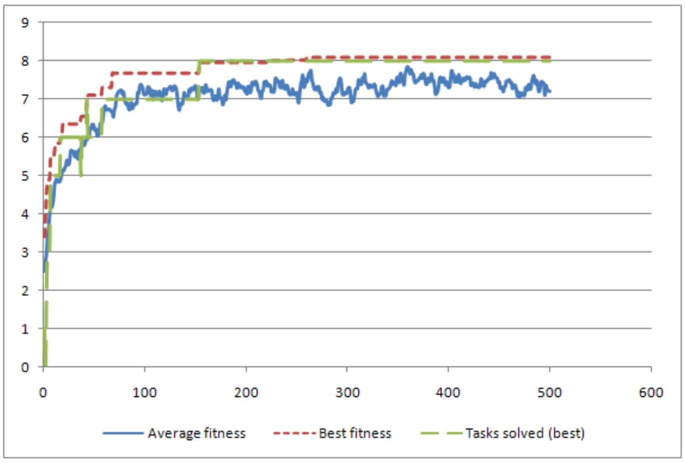


Fig. 4. Algorithm's progress: plot for test IV (row number 4)

The main limitation of our method is its centralization. As we mentioned in the first section, MAS are often distributed (or are at least independent processes or threads). Centralized control must assume that agents are cooperative not competitive. Such an assumption may be acceptable in MAS environments composed of agents belonging to one company. It may be, however, too strong in the case of distributed environments accessible for agents belonging to different companies and having contradictory aims. Therefore we are going to develop a method of coalition formation based on the search of consensus (equilibrium) through negotiation. In such a system the only requirement for agents would be to understand the protocol of communication. Therefore there are no obstacles to make the system fully decentralized and distributed.

## REFERENCES

[1] Jingan Yang and Zhenghu Luo, *Colaition formation mechanism in multi-agent systems based on genetic algorithms*, Applied Computing Soft (Elsevier), 2006.

[2] O. Sheory and S. Kraus, *Task allocation via coalition formation among autonomous agents*, Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 655–661.

[3] G. Weiss—editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, 2000.

[4] J. Arabas, *Wykłady z algorytmw ewolucyjnych*, WNT, 2004.

[5] T. Rahwan, *Algorithms for Coalition Formation in Multi-Agent Systems*, University Of Southampton (PhD), 2007.

[6] C. Tessier—editor, *Conflicting Agents: Conflict Management in Multi-Agent Systems*, Kluwer Academic Publishers, 2001.

[7] M. Wooldridge and N.R. Jennings, *Intelligent Agents: Theory and Practice*, Knowledge Engineering Review, 1995, vol. 10/2, pp. 115–152.

[8] T. Sandholm et al, *Coalition Structure Generation with Worst Case Guarantees*, Artificial Intelligence, vol. 111/(1-2), pp. 209-238, 1999.

[9] B. Horling and V. Lesser, *A survey of multi-agent organizational paradigms*, The Knowledge Engineering Review, vol. 19/4. pp. 281–316, 2005

[10] E. Gamma and J. Vlissides and R. Johnson and R. Helm, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Longman Publishing Co., 1995.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Applying Emphasized Soft Targets for Gaussian Mixture Model Based Classification

Soufiane El Jelali, Abdelouahid Lyhyaoui and Aníbal R. Figueiras-Vidal*
Dept. of Signal Processing and Communications
Univ. Carlos III de Madrid
Av. de la Universidad 30
Leganés, Madrid 28911, Spain
Email: {soufiane, abdel, arfv}@tsc.uc3m.es

***Abstract*—When training machines classifiers, it is possible to replace hard classification targets by their emphasized soft versions so as to reduce the negative effects of using cost functions as approximations to misclassification rates. This emphasis has the same effect as sample editing methods which have proved to be effective for improving classifiers performance. In this paper, we explore the effectiveness of using emphasized soft targets with generative models, such as Gaussian Mixture Models, that offer some advantages with respect to decision (prediction) oriented architectures, such as an easy interpretation and possibilities of dealing with missing values. Simulation results support the usefulness of the proposed approach to get better performance and show a low sensitivity to design parameters selection.**

## I. Introduction

TRAINING machine classifiers with conventional search procedures using hard targets results in minimizing a cost function which is not more than an approximation to the error rate. Fisher type formulations [1] and "decision based" algorithms [2] are also approximations as well as "energy functions" [3], while the Perceptron Rule [4] has convergence difficulties and generalization limitations.

Sample selection or sample editing methods try to compensate the suboptimal character of the conventional approaches by paying more attention to samples that are more important for defining the classification borders. [5] was the first proposal along this line. It was followed by many schemes that were proposed to emphasize sample populations that pay attention to samples lying (approximately) near the borders [6][7][8] or to samples which offer higher errors [9][10]. Some original formulations about boosting schemes to construct classifier ensembles [11][12][13] seem to be rather based on giving importance to erroneous samples, although [14][15] prove that Real Adaboost emphasizes both erroneous samples and those that are near the border; the last reference also propose some generalizations. It is also clear that emphasizing sample populations is very similar to selecting appropiate cost functions. In this sense, the Maximum Margin algorithm used to train Support Vector Machines [16][17][18] can be considered as a procedure which emphasizes both kinds of samples, as well. It is unclear which of these types of samples is more important to get a good design, although the answer seems to be problem dependent [19].

Among other alternatives, the one of substituting the "hard" classification targets for soft versions is interesting, because it leads to an estimation (or regression) problem, in which standard cost functions can be applied without the difficulties mentioned above. This will allow the possibility of directly applying machines that are conceived in order to solve regression problems, such as Gaussian Processes [20][21], for classification tasks. Although there are many forms of generating soft versions of decision targets, such as using convolutional smoothing [22], it seems reasonable to keep the advantages that sample emphasis provides. This was the orientation of [23] and of our previous work [24], in which we demonstrated the effectiveness of using an emphasized combination of the original targets and those provided by an auxiliary classifier to get better classification performance for the most popular family of discriminative (predictive) designs, Multi-Layer Perceptrons (MLP).

In this paper, we extend our studies to generative models, presenting results for the well known Gaussian Mixture Models (GMM), that we apply both in discriminative and soft-target based generative forms to solve classification problems. This study is important not only for the exploration of the general applicability of soft targets approaches, but also because GMM are relatively easy to understand and they accept well principled methods to deal with missing values.

The rest of the paper is organized as follows. In Section 2, we review the basic algorithm that defines the emphasized soft targets. In Section 3, we explain how to solve a classification problem using the soft target definition and GMM models. Section 4 is devoted to check the performance of the proposed approach in a series of experimental settings. We close the paper with the conclusion which emerges from the experiments and some suggestions for further research along this line.

## II. A Method to Construct Emphasized Soft Targets

In order to create soft targets, it is not practical to use the output of a previously trained (auxiliary) classifier, because the

*This work has been partially supported by MEC Pjt. TEC2005-00992. The work of S. El Jelali was also supported by an AECI grant.

decision results of the scheme being trained with these soft targets can be worse than those of the auxiliary machine, because we are trying to repeat its errors. A convex combination of the original (hard) targets and auxiliary machine outputs $\mathrm{o}_{aux}(\mathbf{x})$ can serve to avoid this difficulty; and even more, if we use a well selected local combination parameter $\lambda(\mathbf{x})$ [24], we will have the possibility of applying an emphasis to the samples. A reasonably simple procedure consists on defining soft targets

$$t_s(\mathbf{x}) = \lambda(e(\mathbf{x}))t(\mathbf{x}) + (1 - \lambda(e(\mathbf{x})))\mathrm{o}_{aux}(\mathbf{x}). \qquad (1)$$

where $\mathbf{x}$ is the (training) sample, $t(\mathbf{x})$ is the original (hard) target, $\mathrm{o}_{aux}(\mathbf{x})$ is the output of the auxiliary classifier (in this paper, we will use an MLP for this role), $e(\mathbf{x})$ is the error corresponding to this auxiliary output, and $\lambda(e(\mathbf{x}))$ is a convex combination (emphasis) weight having the form

$$\lambda(e(\mathbf{x})) = \begin{cases} \exp(-\dfrac{(|e(\mathbf{x})| - \mu)^2}{\alpha_1})\text{for} & |e(\mathbf{x})| \leq \mu, \\ \\ \exp(-\dfrac{(|e(\mathbf{x})| - \mu)^2}{\alpha_2})\text{for} & \mu < |e(\mathbf{x})| \leq 2. \end{cases} \qquad (2)$$

$\mu$, $\alpha_1$, $\alpha_2$ being parameters of the Gaussian bells, that must be selected during the design phase.

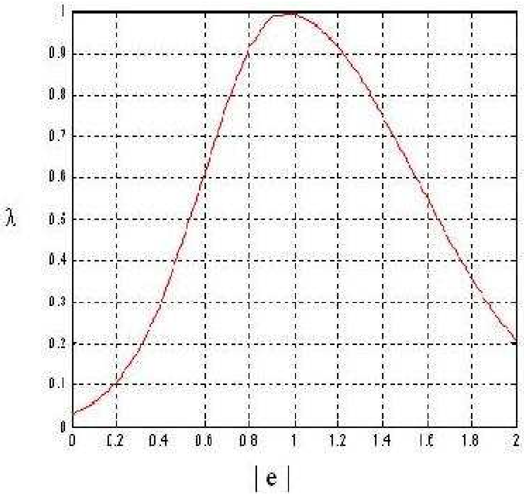


Fig. 1. Form of $\lambda(|e(\mathbf{x})|) : \mu = 1$, $\alpha_1 < \alpha_2$

The form we propose for the convex combination weight is shown in Fig.1. Note that $\mu$ is the value of $|e(\mathbf{x})|$ at which $t_s(\mathbf{x})$ is maximum (unity), and that $\alpha_1$ and $\alpha_2$ control the decay of $t_s(\mathbf{x})$ from this value when samples are clearly well classified ($|e(\mathbf{x})| \to 0$), so they are not very important, or when the samples give highly erroneous results ($|e(\mathbf{x})| \to 2$), in which case they are difficult to classify correctly. So, we can select the value of $|e(\mathbf{x})|$ at which we apply the highest emphasis, and we have flexibility to reduce this emphasis for both "easy" or "impossible" samples.

Although it is obvious that there are many other flexible and reasonable forms for (2), it is easy to verify, as predicted in [25], that the performance of the corresponding designs does not depend critically on the particular emphasis which is employed.

## III. Using GMM

To solve a classification problem by means of GMM models directly, we have to construct these models for both hypotheses, $\widehat{p}(\mathbf{x}|i) = \sum_l \pi_l p_l(\mathbf{x}|i), i = \pm 1$ ( $\{p_l(\mathbf{x}|i)\}$ being Gaussian densities), typically using the Expectation-Maximization (EM) algorithm; then, we construct the approximation to the Maximum A Posteriori (MAP) classifier

$$\widehat{Pr}(1|\mathbf{x}) \underset{-1}{\overset{+1}{\gtrless}} \widehat{Pr}(-1|\mathbf{x})$$

where

$$\widehat{Pr}(i|\mathbf{x}) = \frac{\widehat{p}(\mathbf{x}|i)\widehat{p}(i)}{\sum\limits_{i'} \widehat{p}(\mathbf{x}|i')\widehat{p}(i')}\text{for}i = 1, -1. \qquad (3)$$

$i, i' = 1, -1$; $\{\widehat{p}(i)\}$ can be obtained as relative frequencies.

On the other hand, assuming that we are working with soft target $t_s(\mathbf{x})$, the estimation is addressed under a multidimensional GMM model, first estimating the joint distribution

$$\widehat{p}(t_s, \mathbf{x}) = \sum_l \pi_l p_l(t_s, \mathbf{x}). \qquad (4)$$

with the help of the EM algorithm. After it, since the a posteriori expectation of the target is an estimate of $Pr(1|\mathbf{x}) - Pr(-1|\mathbf{x})$, we look for

$$\begin{aligned} \widehat{t_s}(\mathbf{x}) = \widehat{E}\{t_s|\mathbf{x}\} &= \int t_s \widehat{p}(t_s|\mathbf{x}) dt_s \\ &= \int t_s \frac{\widehat{p}(t_s, \mathbf{x})}{\widehat{p}(\mathbf{x})} dt_s = \sum_l \frac{\pi_l p_l(\mathbf{x})}{\sum\limits_{l'} \pi_{l'} p_{l'}(\mathbf{x})} \int t_s \widehat{p}_l(t_s|\mathbf{x}) dt_s \\ &= \sum_l \frac{\pi_l p_l(\mathbf{x})}{\sum\limits_{l'} \pi_{l'} p_{l'}(\mathbf{x})} \widehat{E}_l\{t_s|\mathbf{x}\}. \end{aligned} \qquad (5)$$

where marginal densities are

$$p_l(\mathbf{x}) = \int p_l(t_s, \mathbf{x}) dt_s \text{for} l = 1, \ldots, L. \qquad (6)$$

It is immediate to check that

$$\widehat{E}_l\{t_s|\mathbf{x}\} = \mathbf{w}_{l,e}^T \mathbf{x}_e = \mathbf{w}_l^T \mathbf{x} + w_{0,l}. \qquad (7)$$

where $\mathbf{w}_l$ is given by the normal equations

$$\mathbf{w}_l = V_{\mathbf{xx},l}^{-1} \mathbf{v}_{t_s\mathbf{x},l}. \qquad (8)$$

$V_{\mathbf{xx},l}$ and $\mathbf{v}_{t_s\mathbf{x},l}$ being the autocovariance matrix of data $\mathbf{x}$ and the cross-covariance vector of target $t_s$ and data $\mathbf{x}$ under (Gaussian) model $p_l(t_s, \mathbf{x})$, that are obtained from the autocovariance matrix of this distribution by deleting its first row and column and as its first column, respectively; finally,

$$w_{0,l} = E_l\{t_s\} - \mathbf{w}_l^T E_l\{\mathbf{x}\}. \qquad (9)$$

where $E_l\{t_s\}$ and $E_l\{\mathbf{x}\}$ are the first element and the remaining vector of the mean vector of $p_l(t_s, \mathbf{x})$, respectively.

Note that $V_{\mathbf{xx},l}$ and $E_l\{\mathbf{x}\}$ are the covariance matrix and mean vector of Gaussian density $p_l(\mathbf{x})$.

The above expressions are formally similar to those corresponding to Mixture of Experts (MoE) ensembles[26][27], as previously mentioned in [28]. However, in our case, we will obtain $\widehat{t}_s(\mathbf{x})$ from the parameters corresponding to the GMM of $p(t_s, \mathbf{x})$ calculated by applying the EM algorithm. Finally, the decision for a new sample $\mathbf{x}_n$ is

$$dec(\mathbf{x}_n) = sgn(\widehat{t}_s(\mathbf{x}_n)). \tag{10}$$

Fig.2 depicts the corresponding decision machine.

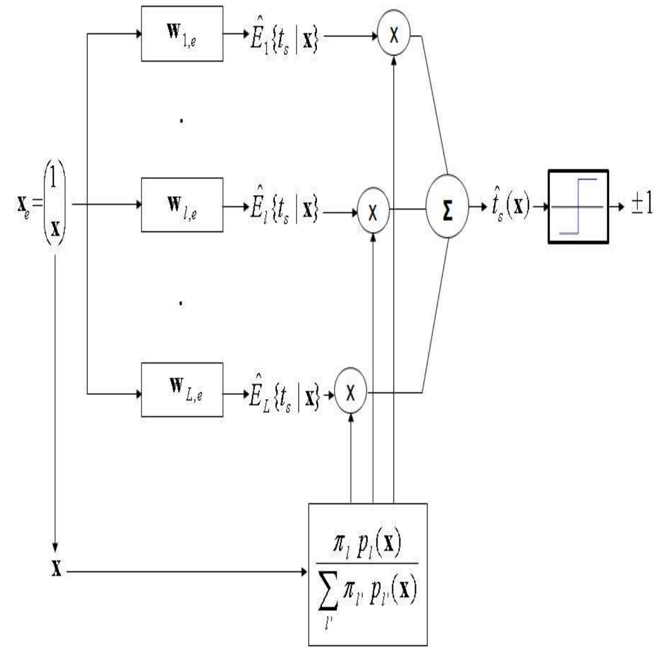

$\mathbf{x}_e = \begin{pmatrix}1\\ \mathbf{x}\end{pmatrix}$
$\mathbf{W}_{1,e}$
$\hat{E}_1\{t_s|\mathbf{x}\}$
$\mathbf{W}_{l,e}$
$\hat{E}_l\{t_s|\mathbf{x}\}$
$\mathbf{W}_{L,e}$
$\hat{E}_L\{t_s|\mathbf{x}\}$
x
x
x
$\Sigma$
$\hat{t}_s(\mathbf{x})$
±1
$\mathbf{x}$
$\frac{\pi_l\, p_l(\mathbf{x})}{\sum_{l'} \pi_{l'}\, p_{l'}(\mathbf{x})}$

Fig. 2. Classification procedure using soft target and GMM models

## IV. Experiments

### A. Datasets

We have tested the proposed method with six datasets that are frequently used as benchmarks for classification problems. The first is the synthetic bidimensional Kwok problem [29]; the other five are taken from the UCI Machine Learning Repository [30]: Abalone (converted into a binary problem according to [31]), Breast Cancer, Contraceptive, Ionosfera, and Pima Indian. We will refer to them as kwo, aba, bre, con, ion, and pim, respectively. Table 1 shows their main characteristics.

### B. Training

All data were normalized -if not previously done- between -1 and 1.

We use an MLP with $N$ hidden neurons as the auxiliary machine. The number of hidden neurons, as well as parameters $\mu$, $\alpha_1$ and $\alpha_2$, and the number of components of GMM, $L$ for the joint model and $L_1$, $L_{-1}$ for the separate class models, are found by means of a 10-fold cross-validation (CV) using 10 runs, exploring the following values:

-$N$: 4, 6, 8, 10 ,12, 14, 16

-$\mu$: 0.01, 0.1, 0.3, 0.6, 1, 1.2, 1.6, 2

-$\alpha_1$, $\alpha_2$: 0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 3, 4, 5

-$L$: 4, 5, 6, 7, 8, 9, 10

-$L_1$, $L_{-1}$: 2, 3, 4, 5

TABLE I
Characteristics of the test problems

| **Dataset** | **Train** $(\mathbf{C_{+1}/C_{-1}})$ | **Test** $(\mathbf{C_{+1}/C_{-1}})$ | $\#\mathbf{dim}$ |
|---|---|---|---|
| **kw** | 500<br>(200/300) | 10200<br>(4080/6120) | 2 |
| **aba** | 2507<br>(1238/1269) | 1670<br>(843/827) | 8 |
| **bre** | 420<br>(145/275) | 279<br>(96/183) | 9 |
| **con** | 883<br>(506/377) | 590<br>(338/252) | 9 |
| **ion** | 201<br>(101/100) | 150<br>(124/26) | 34 |
| **pim** | 461<br>(161/300) | 307<br>(107/200) | 8 |

These experiments did not present important difficulties due to eventual closeness to singularity of the covariance matrices [32]; when this appears in other cases, the solution proposed in [33] can be applied.

Table 2 presents the results of the experiments, as well as the values of the design parameters that are obtained from applying CV. With respect to direct MLP and MAP GMM designs, the results of our approach (SOFT GMM) show an advantage going from very small for kwo and bre to important for pim, ion, and con. The results are also competitive with those provided by an SVM with a Gaussian kernel whose parameters have also been optimized by a 10-fold CV process (penalty factor $C$ : $[10^{-1}11010^{2}10^{3}10^{4}]$), taking the kernel dispersion as $\sigma = \sqrt{\#dim/2}$. We have used the IRWLS SVM toolbox for Matlab [37] for pattern recognition, with tolerance $\epsilon = 10^{-5}$ to provide the accuracy required for the solution of the Quadratic Programming (QP) algorithm.

To make evident the advantage of the proposed method, Fig. 3 shows the classification boundaries for kwo obtained with the optimal MLP, the MAP-GMM method, and the proposed approach, compared with the theoretical boundary. Note that the greater proximity of the proposed classifier boundary to the optimal boundary means that it correctly classifies some samples that MLP and MAP-GMM place in the wrong decision region.

TABLE II
AVERAGED PERCENTAGES OF CORRECT CLASSIFICATION ± STANDARD DEVIATION AND DESIGN PARAMETERS FOR EACH METHOD (MLP: $N$; MAP GMM: $k_1, k_2$; SOFT GMM: $k, N, \mu, \alpha_1, \alpha_2$; AND SVM: $C$) FOR THE TEST DATASETS WITH A 10-FOLD CROSS-VALIDATION

| Dataset | MLP | MAP GMM | SOFT GMM | SVM |
|---|---|---|---|---|
| **kwo** | $83.49$<br>$\pm 3.32$<br>$N = 16$ | $84.77$<br>$\pm 0.59$<br>$k_1 = 3,$<br>$k_2 = 3$ | $85.00$<br>$\pm 0.83$<br>$k = 9,$<br>$N = 6,$<br>$\mu = 0.3,$<br>$\alpha_1 = 0.01$<br>$\alpha_2 = 0.1$ | $83.68$<br>$\pm 0.59$<br>$C = 10^3,$<br>$\sigma = 1$ |
| **aba** | $78.12$<br>$\pm 0.54$<br>$N = 12$ | $72.90$<br>$\pm 0.84$<br>$k_1 = 4,$<br>$k_2 = 4,$ | $73.01$<br>$\pm 0.67$<br>$k = 9,$<br>$N = 12,$<br>$\mu = 1.2,$<br>$\alpha_1 = 0.1$<br>$\alpha_2 = 0.1$ | $77.47$<br>$\pm 4.44$<br>$C = 10,$<br>$\sigma = 2$ |
| **bre** | $97.51$<br>$\pm 0.60$<br>$N = 8$ | $94.90$<br>$\pm 1.46$<br>$k_1 = 3,$<br>$k_2 = 4$ | $95.34$<br>$\pm 0.91$<br>$k = 8,$<br>$N = 10,$<br>$\mu = 0.3,$<br>$\alpha_1 = 0.01,$<br>$\alpha_2 = 0.05$ | $97.49$<br>$\pm 0.34$<br>$C = 1,$<br>$\sigma = 2.12$ |
| **con** | $70.38$<br>$\pm 2.24$<br>$N = 10$ | $62.88$<br>$\pm 1.41$<br>$k_1 = 4,$<br>$k_2 = 4$ | $65.69$<br>$\pm 1.98$<br>$k = 8,$<br>$N = 6,$<br>$\mu = 1.6$<br>$\alpha_1 = 0.1,$<br>$\alpha_2 = 4$ | $70.39$<br>$\pm 0.65$<br>$C = 10,$<br>$\sigma = 2.12$ |
| **ion** | $93.22$<br>$\pm 1.48$<br>$N = 6$ | $93.15$<br>$\pm 2.56$<br>$k_1 = 3,$<br>$k_2 = 3$ | $94.52$<br>$\pm 1.82$<br>$k = 9,$<br>$N = 6,$<br>$\mu = 1.2,$<br>$\alpha_1 = 0.05$<br>$\alpha_2 = 0.5$ | $97.80$<br>$\pm 0.45$<br>$C = 10,$<br>$\sigma = 4.12$ |
| **pim** | $78.33$<br>$\pm 1.33$<br>$N = 6$ | $72.72$<br>$\pm 1.40$<br>$k_1 = 2,$<br>$k_2 = 2$ | $77.37$<br>$\pm 1.50$<br>$k = 6,$<br>$N = 4,$<br>$\mu = 0.3,$<br>$\alpha_1 = 0.01,$<br>$\alpha_2 = 1$ | $75.44$<br>$\pm 0.79$<br>$C = 10^2,$<br>$\sigma = 2.12$ |

Of course, this advantage is obtained at a high training cost, since we need to select the best of $\#N \times \#\mu \times \#\alpha_1 \times \#\alpha_2 \times \#L$ designs, and not just $\#N$ (for a single MLP), $\#L_1 \times \#L_{-1}$ (for the MAP GMM case), or $\#C$ (for SVM schemes). However, once the proposed machine is trained, its application requires a pretty moderate computational effort, equivalent to that of using a GMM for regression.

An additional problem could appear in terms of sensitivity with respect to the values of the design parameters selected by CV. To explore this in a exhaustive form (considering variations of all the design parameters around each optimal design) is too complex. A more reasonable option is to present the results of the so-called “omniscient” designs, those that select the design parameters according to their performances for the test sets. Of course, these omniscient versions are not acceptable as valid designs, but comparing their performances (and the corresponding design parameter values) with those of the optimal CV designs is an indication of the sensitivity of these machines.

TABLE III
“OMNISCIENT” RESULTS: AVERAGED PERCENTAGES OF CORRECT CLASSIFICATION (± STANDARD DEVIATION) OF MLP, MAP GMM, SOFT GMM, AND SVM FOR THE TEST DATASETS. DESIGN PARAMETERS ARE ALSO INDICATED

| Dataset | MLP | MAP GMM | SOFT GMM | SVM |
|---|---|---|---|---|
| **kwo** | $84.30$<br>$\pm 0.66$<br>$N = 6$ | $85.30$<br>$\pm 0.62$<br>$k_1 = 2,$<br>$k_2 = 5$ | $85.31$<br>$\pm 1.06$<br>$k = 6,$<br>$N = 12,$<br>$\mu = 0.3,$<br>$\alpha_1 = 0.1$<br>$\alpha_2 = 0.01$ | $83.68$<br>$\pm 0.59$<br>$C = 10^3,$<br>$\sigma = 1$ |
| **aba** | $78.20$<br>$\pm 0.41$<br>$N = 6$ | $72.97$<br>$\pm 0.58$<br>$k_1 = 5,$<br>$k_2 = 5$ | $73.01$<br>$\pm 0.67$<br>$k = 9,$<br>$N = 12,$<br>$\mu = 1.2,$<br>$\alpha_1 = 0.1,$<br>$\alpha_2 = 0.1$ | $77.42$<br>$\pm 4.44$<br>$C = 10,$<br>$\sigma = 2$ |
| **bre** | $97.53$<br>$\pm 0.56$<br>$N = 4$ | $95.94$<br>$\pm 1.08$<br>$k_1 = 5,$<br>$k_2 = 4$ | $95.77$<br>$\pm 1.10$<br>$k = 6,$<br>$N = 10,$<br>$\mu = 0.3,$<br>$\alpha_1 = 0.001,$<br>$\alpha_2 = 1.5$ | $97.63$<br>$\pm 0.39$<br>$C = 10,$<br>$\sigma = 2.12$ |
| **con** | $70.51$<br>$\pm 2.24$<br>$N = 6$ | $63.15$<br>$\pm 0.95$<br>$k_1 = 2,$<br>$k_2 = 2$ | $65.69$<br>$\pm 1.98$<br>$k = 8,$<br>$N = 6,$<br>$\mu = 1.6,$<br>$\alpha_1 = 0.1,$<br>$\alpha_2 = 4$ | $71.07$<br>$\pm 0.65$<br>$C = 10^2,$<br>$\sigma = 2.12$ |
| **ion** | $93.22$<br>$\pm 1.48$<br>$N = 6$ | $93.15$<br>$\pm 2.56$<br>$k_1 = 3,$<br>$k_2 = 3$ | $94.54$<br>$\pm 1.89$<br>$k = 10,$<br>$N = 8,$<br>$\mu = 1,$<br>$\alpha_1 = 5,$<br>$\alpha_2 = 2$ | $97.80$<br>$\pm 0.45$<br>$C = 10,$<br>$\sigma = 4.12$ |
| **pim** | $78.80$<br>$\pm 0.97$<br>$N = 12$ | $72.89$<br>$\pm 1.80$<br>$k_1 = 2,$<br>$k_2 = 3$ | $78.92$<br>$\pm 0.98$<br>$k = 6,$<br>$N = 12,$<br>$\mu = 0.3,$<br>$\alpha_1 = 0.1,$<br>$\alpha_2 = 0.01$ | $78.93$<br>$\pm 0.56$<br>$C = 1,$<br>$\sigma = 2.12$ |

Table 3 presents the same values than Table 2, but for the corresponding omniscient machines. Note that, even having to select much more design parameters, the proposed method

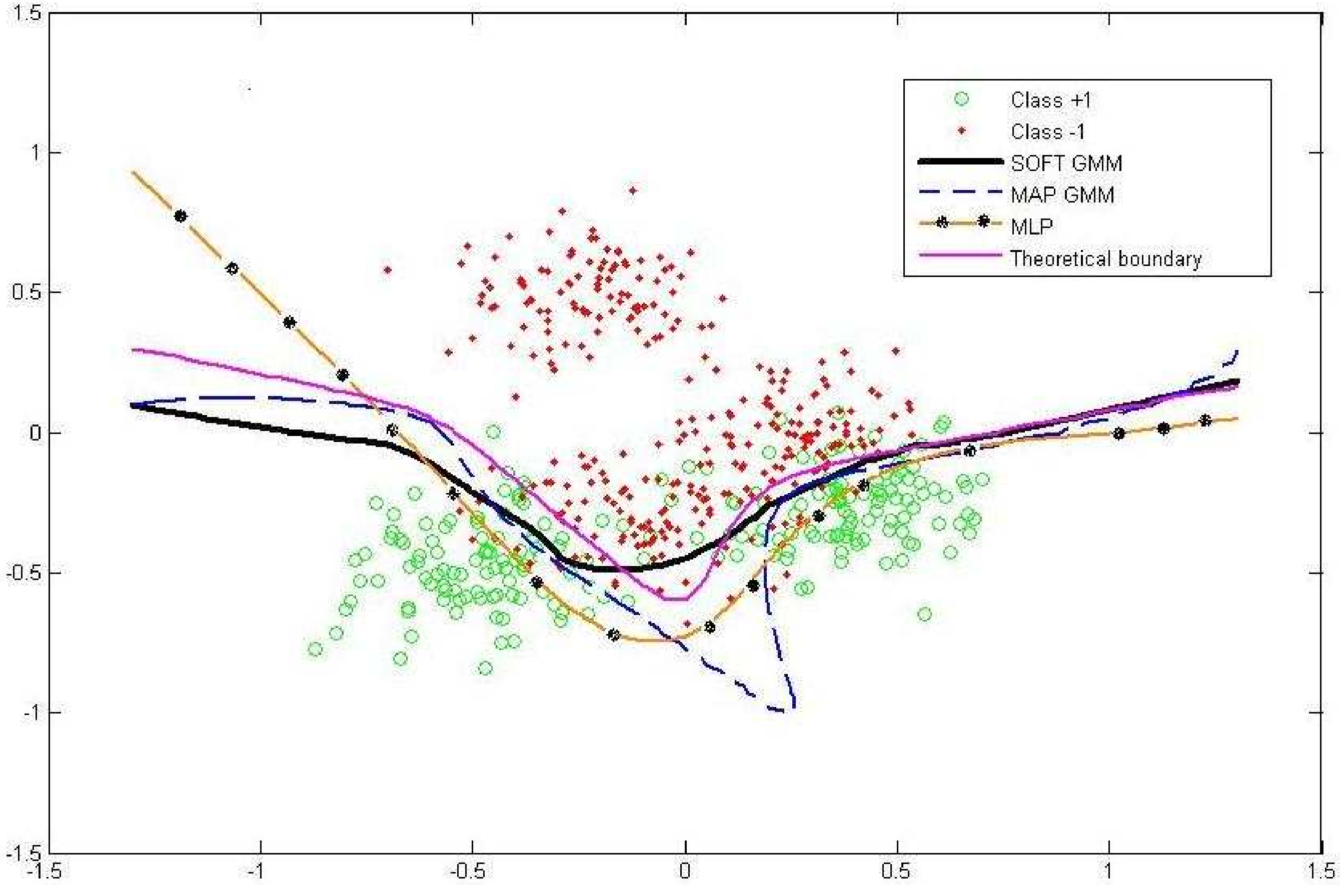


Fig. 3. Boundaries for the three methods mentioned in Table 2 compared with the theoretical boundary for test problem kwo, which is subsampled to 500 samples ($C_{+1} : 200/C_{-1} : 300$)

gets the omniscient result in two problems (aba and con), while the MAP GMM gets it just once (ion). Additionally, even with different design parameters, the CV and omniscient designs for our method show small performance differences for kwo and bre, and significant differences only for pim, just the problem for which the improvement of our proposal with respect to standard designs is the highest. When applying MAP GMM schemes, small differences also appear for kwo and con, and are important for bre. For MLPs, CV offers the omniscient design in ion, and a significant difference in kwo and pim, even $N$ being the only design parameter to be selected. So, it appears that we have a reduced sensitivity when using our designs.

## V. Conclusion

In this work, we have extended the idea of emphasized target smoothing to GMM based decision (it is also possible to extend it to other generative models, such as Parzen windows [34], using Nadaraya-Watson regression [35][36] to deal with soft targets; similar results are obtained). As in the previously studied case of MLP, extensive simulation results allow to conclude that the proposed approach does allow eventual performance improvements with respect to the component standard machines, and it even offers a low sensitivity with respect to the selection of design parameters values. Obviously there is a need for more computational effort to design the proposed classifiers, but their operation is computationally light, and, in the case of being applied to generative models, they have the advantages of allowing a relatively easy interpretation and make it possible to apply principled methods to deal with missing values.

Further research on the extensions of the proposed method will address its use in GP machines, that do not have direct forms for classification purposes because they are based on interpreting targets as Gaussian processes defined on the observations domain.

## References

[1] R. A. Fisher, “The use of multiple measurements in taxonomic problems”, Annals of Eugenics, **7**, Pt. II, 179–188, 1936.

[2] S. Y. Kung, J. S. Taur, “Decision-based neural networks with signal/image classification applications”, IEEE Trans. Neural Networks, **6**, 170–181, 1995.

[3] B. A. Telfer, H. H. Szu, "Energy functions for minimizing misclassification error with minimum-complexity networks", Neural Networks, **7**, 809–818, 1994.

[4] F. Rosenblatt, "The Perceptron: A probabilistic model for information storage and organization in the brain", Psychological Review, **65**, 386–408, 1958.

[5] P. E. Hart, "The condensed nearest neighbor rule", IEEE Trans. Information Theory, **14**, 515–516, 1968.

[6] J. Sklansky, L. Michelotti, "Locally trained piecewise linear classifiers", IEEE Trans. Pattern Anal. Machine Intelligence, **2**, 101–111, 1980.

[7] M. Plutowski, H. White, "Selecting concise training sets from clean data", IEEE Trans. Neural Networks, **4**, 305–318, 1993.

[8] S. H. Choi, P. Rockett, "The training of neural classifiers with condensed datasets", IEEE Trans. Sys., Man, and Cybernetics, Pt. B, **32**, 202–206, 2002.

[9] P.W. Munro, "Repeat until bored: A pattern selection strategy", Adv. in Neural Inf. Proc. Sys. 4 (J. E. Moody et al., eds.), 1001–1008; San Mateo, CA: Morgan Kaufmann, 1992.

[10] C. Cachin, "Pedagogical pattern selection strategies", Neural Networks, **7**, 171–181, 1994.

[11] Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm", Proc. 13th Intl. Conf. Machine Learning, 148–156; Bari (Italy), 1996.

[12] Y. Freund, R. E. Schapire "Game theory, on–line prediction, and boosting", Proc. 9th Annual Conf. on Comput. Learning Theory, 325–332; Desenzano di Garda (Italy), 1996.

[13] R. E. Schapire, Y. Singer, "Improved boosting algorithms using confidence-rated predictions", Machine Learning, **37**, 297–336, 1999.

[14] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas- García, A. R. Figueiras-Vidal, "Boosting by weighting critical and erroneous samples", Neurocomputing, **69**, 679–685, 2006.

[15] V. Gómez-Verdejo, J. Arenas-García, A. R. Figueiras-Vidal, "A dynamically adjusted mixed emphasis method for building boosting ensembles", IEEE Trans. Neural Networks, **19**, 3–17, 2008.

[16] B. E. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers", Proc. 5th Annual Workshop Comp. Learning Theory (D. Hassler, ed.), 144–152; Pittsburgh, PA: ACM Press, 1992.

[17] C. Cortes, V. Vapnik, "Support Vector networks", Machine Learning, **20**, 273–297, 1995.

[18] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf "An introduction to kernel–based learning algorithms", IEEE Trans. Neural Networks, **12**, 181–201, 2001.

[19] L. Franco, S. A. Cannas, "Generalization and selection of examples in feed–forward neural networks", Neural Computation, **12**, 2405–2426, 2000.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[21] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press, 2006.

[22] R. Reed, S. Oh, and R. J. Marks, II, "Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter", IEEE Trans. Neural Networks, **6**, 529–538, 1995.

[23] D. Gorse, A. J. Shepperd, J. G. Taylor, "The new ERA in supervised learning", Neural Networks, **10**, 343–352, 1997.

[24] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "An emphasized target smoothing procedure to improve MLP classifiers performance", Proc. 16th European Symp. Artificial Neural Networks, 499–504; Bruges (Belgium), 2008.

[25] L. Breiman, "Combining predictors", *in Combining Artificial Neural Nets.: Ensemble and Modular Multi-net Systems* (A. J. C. Sharkey, ed.), 31–50; London, UK: Springer, 1999.

[26] R. A. Jacobs, M. I. Jordan, "A competitive modular connectionist architecture", in Advances in Neural Info. Proc. Sys. 5 (D. Touretzky, ed.), 767–773; San Mateo, CA: Morgan Kaufmann, 1991.

[27] M. I. Jordan, R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM algorithm", Neural Computation, **6**, 181–214, 1994.

[28] L. Xu, M.I. Jordan, G. E. Hinton, "An alternative model for Mixtures of Experts", in Advances in Neural Information Processing Systems 7, 633–640. MIT Press, 1995.

[29] J. T. Kwok, "Moderating the output of Support Vector classifiers", IEEE Trans. Neural Networks, **10**, 1018–1031, 1999.

[30] C. L. Blake, C. J. Merty: UCI Repository of Machine Learning Databases: www.ics.uci.edu/∼mlearn

[31] A. Ruiz, P. E. López-de-Teruel, "Nonlinear kernel-based statistical pattern analysis", IEEE Trans. Neural Networks, **12**, 16–32, 2001.

[32] C. Archambeau, J. A. Lee, M. Verleysen, "On convergence problems of the EM algorithm for finite mixture models", Proc. 11th European Symposium on Artificial Neural Networks, 99–106; Bruges (Belgium), 2003.

[33] C. Archambeau, F. Vrins, M. Verleysen, "Flexible and robust Bayesian classification by finite mixture models", Proc. 12th European Symposium on Artificial Neural Networks, 75–80; Bruges (Belgium), 2004.

[34] É. A. Nadaraya, "On estimating regression", Theory of Probability and Its Applications, **9**, 141–412, 1964.

[35] G. S. Watson, "Smooth regression analysis", Sankhyā: The Indian Journal of Statistics, Series A, **26**, 259–279, 1964.

[36] E. Parzen, "On estimation of a probability density function and mode", Annals of Math. Statistics, **33**, 1065–1076, 1962.

[37] F. Pérez–Cruz, "IRWLS Matlab toolbox to solve the SVM for pattern recognition and regression estimation", 2002. Available: http://www.tsc.uc3m.es/∼fernando/

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Improving Naïve Bayes Models of Insurance Risk by Unsupervised Classification

Anna Jurek
Institute of Mathematics
Technical University of Lodz
Wolczanska 215, 90-924 Lodz, Poland
Email: ankajurek@poczta.fm

Danuta Zakrzewska
Institute of Computer Science
Technical University of Lodz
Wolczanska 215, 90-924 Lodz, Poland
Email: dzakrz@ics.p.lodz.pl

***Abstract*—In the paper application of Naïve Bayes model, for evaluation of the risk connected with life insurance of customers, is considered. Clients are classified into groups of different insurance risk levels. There is proposed to improve the efficiency of classification by using cluster analysis in the preprocessing phase. Experiments showed that, however the percentage of correctly qualified instances is satisfactory in case of Naïve Bayes classification, but the use of cluster analysis and building separate models for different groups of clients improve significantly the accuracy of classification. Finally, there is discussed increasing of efficiency by using cluster validation techniques or tolerance threshold that enables obtaining clusters of very good quality.**

## I. Introduction

RISK management is one of the most important area of interests in insurance industry. All the activities of insurance companies are usually connected with risk of policyholders claiming for big damages and the necessity of paying them a large amount of money. The level of the risk became the crucial factor influencing insurance company profits and may even decide on prosperity or failure in the industry. Evaluation of the risk level plays an important role in defining insurance companies' strategies. Having data of all the clients together with the historical information of damage claiming, insurers can estimate risk connected with the sale of every insurance policy. Building risk evaluation models may help them to make the decision concerning acceptance or rejection of new insurance applications. There exist different analysis methods that may support that process, as the most important there should be mentioned: predictive modeling, risk modeling, scoring or risk level qualification.

In the paper there is considered using of data mining techniques to classify life insurance applications as good or bad from the point of view of risk undertaken by the company. By examining historical data of insurance company, clients with certain characteristic features, may be qualified into the groups of different risk level. We investigate application of Naïve Bayes classifier for predicting the risk and for assigning every new customer into the proper group. Naïve Bayes algorithm is a very simple tool for machine learning and data mining. Despite of the fact that the conditional assumption, on which it is based, is rarely fulfilled in the real world, it ensures high accuracy of obtained results. The superb performance of its classification effects was explained in [1]. We propose to increase the efficiency of the classification by using clustering techniques in the preprocessing phase. The experiments showed that, however results obtained by building Naïve Bayes models, measured by the percentage of correctly qualified instances, were satisfactory, but application of clustering algorithm in the preprocessing stage improved significantly the accuracy of the classification. What is more using cluster validity measurement techniques enables choosing of optimal clustering schema and the further increase of correctness of classification.

The paper is organized as follows. In the next section some research concerning application of data mining algorithms in insurance industry is presented. In Section 3, data preparation process as well as applied algorithms are introduced. Then in Section 4 experiments and their results are described. Section 5 presents discussion and evaluation of obtained results. Finally some concluding remarks and future work are proposed.

## II. Relevant Research

Analysis of large historical data sets has been used in actuarial investigations that concerns making optimal underwriting and price decisions for many years [2]. Development of data mining techniques enabled to increase an effectiveness of data analysis in insurance area. The main research focus on building predictive and risk evaluation models, that allow for identifying customers' behaviours and supporting insurance decision making. Especially data mining techniques were examined for fraud detection, discovering insurance risk or ameliorating customer services (see [3], [4], [5] for example). Examination of different application areas and their technical challenges are presented in [6]. In [7], author described, how the financial situation of the insurance company, may be improved by using such techniques as: decision trees, generalized linear modeling or logistic regression. In [5], there is considered effectiveness of association rules and neural segmentation for analyzing large data sets collected in health insurance information system. Authors considered detecting patterns in ordering pathology services as well as classifying practitionners, and they concluded that data mining techniques allow to obtain the results that are not achievable by conventional methods. Classification trees were examined in the Worker Compensation insurances area in [8], where CART algorithm [9] was used

978-83-60810-14-9/08/$25.00 © 2008 IEEE

to build a model that classifies each claim as “likely to become more serious” or “not to become more serious”. Comparison of examined algorithm with logistic regression showed that CART model guarantees better classification results. In the same paper, there were also considered classification trees and hybrid modeling for building predictive models of hospital costs in health insurance. Decision trees were also used, in [10], to generate a predictive model, that may help insurance companies in the identification of claims which are the most likely to generate cost savings.

Data mining techniques were also investigated in fraud detection in health insurance area, taking into account such methods as k-Nearest Neighbor [11], multi-layer perceptron network [12] as well as heuristics and machine learning [13]. Broad review of application of neural networks, fuzzy logic and genetic algorithms in insurance industry can be found in [14]. An overview of fuzzy logic applications in insurance is presented in [15], where two approaches were studied: fuzzy logic applied separately and its combination with neural networks and genetic algorithms. The author reported application of fuzzy logic techniques in such insurance areas as classification, underwriting, projected liabilities, pricing, asset allocation, and investments. As the main fuzzy logic tool used in risk and claim classification, there was presented c-means clustering [15], which in turn was recognized, by Derrig and Ostaszewski, who classified insurance claim depending on level of fraud, as valuable addition to other methods but not the best technique (see [16] for example) to use separately.

Performance of Naïve Bayes models was mainly investigated in fraud detection area (see [17], [18]). Many empirical comparisons between Naïve Bayes technique and modern decision trees: C4.5 and C4.4 as well as Support Vector Machine, showed that Naïve Bayes predicts equally well [19]. In comparative study presented in [17], authors concluded that smoothed Naïve Bayes gave better results than C4.5 decision trees in automobile fraud insurance claims. Experimental study in individual disability income insurance fraud detection [20] showed that Naïve Bayes predictive models outperformed decision tree and Multiple Criteria Linear Programming models in terms of classification accuracy.

The accuracy of Naïve Bayes classifier was improved by combining it with other methods in several papers (see [21], [22], [23], [24] for example). Naïve Bayes classifier was enhanced by application of features selection methods, discretization or using boosting procedures. In [21], authors introduced Selective Bayesian Classifier (SBC), which uses only attributes not removed by C4.5 decision trees. They run C4.5 algorithm on the 10% samples of the all data shuffled together. That action was repeated five times, then the set of attributes was built as an union of those appearing only in the first three levels of the simplified decision trees. Created that way set of selected attributes was taken into account by Naïve Bayes classifier. Experiments conducted on the ten data sets showed that SBC learns faster than Naïve Bayes classifier on all the data sets, and the obtained results were improved up to 7.9%.

In [23], there were used together: dicretization, feature selection and boosting procedures, as techniques for Naïve Bayes improvements. Discretization of attributes with continuous values, was done by applying entropy-based method [25]. Features were selected by using filter that computes empirical mutual information between features and discard low-valued features, measured by gain ratio. The gain ratio value of selected attributes were to exceed a fixed threshold. In the proposed algorithm there was included boosting technique Adaboost introduced in [26]. During each iteration of the algorithm, there is applied entropy discretization technique and redundant attributes are removed by using the gain ratio feature selection method. Such defined algorithm is not as easily interpretable as simple Bayes model, but much more comprehensible than neural networks for example. Experiments conducted on 26 data sets showed that the proposed method was more accurate than Bayes algorithm in 12 cases, with the average error rate of about 20% less than that of simple Bayes algorithm. Another improvement of Naïve Bayes model, called hidden Naïve Bayes was proposed in [24]. The main idea of the proposed model consists in creating attribute hidden parents, that represent influence of all the other attributes. Experiments done on 36 data sets showed that considered algorithm outperforms Naïve Bayes, SBC as well as C4.5 decision trees.

## III. Methodology

In considered model risk evaluation is based on classification rules, that may be built by exploring historical data collected in daily activity of insurance companies. Having characteristic features of all customers together with history of their policies and their claims, insurers can determine groups of different risk level. In our research, three groups of clients are distinguished. The first one, containing customers of low insurance risk (the best clients), the second one of medium risk level and the third group, that may be characterized by high risk (clients that should be avoided). The main idea consists in classifying each of new customers to one of the groups to predict the insurance risk. On the basis of information about the potential client, insurers can almost automatically estimate risk and refuse or accept potential customer application.

In our investigations two different approaches are considered: in the first one classification is based on all the data contained in the database; in the second one customers’ data are divided into clusters of certain similarities and different classification rules are built for each segment separately. In that approach the new client is firstly assigned to one of the groups by unsupervised classification and then, the decision rule appriopriate to considered cluster is applied.

### A. Data preparation

In the present study, we will limit our research into life insurance risk evaluation. However, insurance companies are very interested in finding effective risk evaluation models, but they are very sensitive and reluctant in sharing their data publicly, they do not allow for using their data even

TABLE I
ATTRIBUTES OF LIFE INSURANCE DATABASE

| No | Attribute name | Attribute type |
|---|---|---|
| 1 | Sex | qualitative |
| 2 | Profession | qualitative |
| 3 | Region | qualitative |
| 4 | Hobby | qualitative |
| 5 | Drinking alcohol | qualitative |
| 6 | Smoking | qualitative |
| 7 | Disease | qualitative |
| 8 | Weight | quantitative |
| 9 | Age | quantitative |
| 10 | Blood pressure | quantitative |
| 11 | Maritial status | qualitative |

for research goal. Problems connected with that fact, and its consequences were described with details in [10], where the author recognized lack of the access to real data as the main reason of limitations in understanding data mining and predictive modeling in insurance area.

We base our research on artificially generated data sets, with different attributes that may play a crucial role in profitability of life insurance. Considered client features, are strictly connected with medical exams that customers are obliged to fulfill. All the attributes, used in the research represent information required in medical exam of Swiss Life Insurance and Pension Company [27]. Some exemplary tests, containing similar questions, may be also found at [28]. All the attributes are presented in Table I. Most of them are of qualitative type, except of the following: "weight", "age" and "blood pressure".

For the purpose of Naïve Bayes classification all numerical data should be dicretized into ranges partitioned into intervals. In case of the attribute "weight" three ranges defined as underweight, proper weight and overweight, may be distinguished; in case of attribute "age" values may be binned into five ranges and for "blood pressure" we may have three: low, normal and high. Categorical data, in turn, should be binned into meta - classes (for example: region instead of city). This operation is necessary because Naïve Bayes model relies on calculating probability and cardinality of values should be reduced. Since all attributes are used in the classification process, all of them should be binned.

### B. Naïve Bayes classifier

Naïve Bayes models are the simplest forms of Bayesian network for general probability estimation, detailed description of their functionality was presented in [29]. Naïve Bayes classifier, assumes conditional independence of input data. This is the strong assumption, that seems to be unrealistic in the insurance domain, but a history of empirical studies shows that even in such cases the method presented good performance [18].

By application of Naïve Bayesian algorithm, we obtain probability distribution of belonging into classes. In uncertain cases objects may not be assigned into any group (reject option) or similarly to fuzzy logic techniques may be allocated into more than one class. Other advantages of classifier probabilistic output, such as changing utility functions, compensating for class imbalance or combining models, were described in [30]. Naïve Bayesian algorithm also naturally deals with missing values, what is difficult to achieve by decision trees or neural networks methods. What is more, obtained models are very easy to understand, without further investigations, that feature is not valid for all methods, to mention neural networks as an example. Comparisons of the performance of Naïve Bayes classifier and other classification techniques, like decision trees, neural networks, kNN, Support Vector Machine or rule-learners, were presented in [31]. As main features, which allow Naïve Bayes technique to outperform other algorithms, there were mentioned: speed of learning and classification, tolerance to missing values, explanation ability as well as model parameter handling. To increase of the classification accuracy, the author suggested application of ensemble methods [31].

Naïve Bayes model is based on maximum likelihood, that uses very well known Bayes' formula:

$$P(H_j/A) = \frac{P(A/H_j)P(H_j)}{\sum_{i=1}^{n} P(A/H_i)P(H_i)}, \tag{1}$$

where $j \in 1...n$,$P(A/H_j)$ means conditional probability and is defined as:

$$P(A/H_j) = \frac{P(A \cap H_j)}{P(H_j)} \tag{2}$$

$H_j$ is an event, that means belonging to the group of risk. $A$ is a vector of customer attributes. $P(H_j/A)$ is the probability that person described by $A$ belongs to $H_j$; $P(H_j)$ means probability of belonging to group $H_j$. $P(A/H_j)$ is the probability that customer from $H_j$ is described by $A$.

Algorithm compares features' vector of a new customer with all records in the database and computes the probability of memberships of all of the groups, then the considered client is assigned into the group of risk, for which probability of belonging is the highest. Efficiency of the algorithm may be evaluated by checking assignements for test set of customers' data, by comparing of obtained results with real belongings to the group of certain risk level. Assessment of classification accuracy is done by calculating the percentage of correctly classified records. Addidtional advantage of the technique is that it does not have to use all attributes and can work just with few selected ones. The choice of the attributes that guarantee the most accurate results may be done by the use of training data sets in the preprocessing phase.

### C. Clustering

Cluster analysis techniques become very popular in customer segmentation area. One of the main advantage of the clustering technique is that it does not assume any specific distribution on the data. The main disadvantage of the method is the high dependence of experts' opinions in many cases. There exist many clustering techniques that may be used in customer grouping, the broad review of the most popular of them is presented in [32]. In the current research, k-means algorithm has been chosen, because of its simplicity and efficiency. This algorithm performed very well in experiments concerning credit risk scoring [33]. However the method depends significantly on the initial assignments, what may entail in not finding the most optimal cluster allocation at the end of the process, but k-means is very efficient for large multidimensional data sets [32].

Segmentation is done according to attributes that may play the most crucial role in life insurance: “Drinking alcohol”, “Smoking”, “Profession”, for different number of required clusters. The distance between two objects is measured by Manhattan function. As optimal clustering schema may differ depending on data sets, validity measurement techniques are used to evaluate obtained clusters and to make the best choice for their numbers.

Different cluster validity indices were introduced to examine quality of grouping results. All of them and their application for different clustering algorithms are broadly investigated in [34]. To find out the best clustering schema, it has been chosen Davies-Bouldin (DB) index because of its simplicty. It is based on dispersion and cluster distance measures, which means taking into account: internal variance and external similarity. Low $DB$ value means that clusters are compact and well separated. $DB$ is defined as follows [34]:

$$DB = \frac{1}{k}\sum_{i=1}^{k} R_i, \tag{3}$$

where $k$ is the number of clusters, $R_i$ is the ratio of dispersion measures and distance of clusters $C_i$ and $C_j$: $R_i = max\left\{R_{ij}, i \neq j\right\}, R_{ij} = (S_i + S_j)/D_{ij}$. $D_{ij}$ is measured by the distance between centroids $\nu_i$ and $\nu_j$ of $C_i$ and $C_j$ respectively, and dispersion measure of cluster $C_i$ is defined by:

$$S_i = \left(\frac{1}{|C_i|}\sum_{x \in C_i} d^p(x, \nu_i)\right)^{1/p}, p > 0, \tag{4}$$

where $d(x, y)$ is the distance between $x$ and $y$.

## IV. EXPERIMENTS

Experiments were done on artificially generated data sets, with attributes presented in Table I. Two databases were filled with values randomly. The first one contains five hundred records with eight attributes: “sex”, “profession”, “region”, “hobby”, “drinking alcohol”, “smoking”, “disease” and “weight”. The second one, of eight hundred records is characterized by eleven attributes, all of them are presented in Table I. The main aim of the experiments was to examine how the use of clustering techniques in the preprocessing stage may influence classification accuracy. There were compared results of classification done by using all the records of considered data sets, with effects obtained on segments of similar customers. It was also examined how the quality of results changes depending on choice of attributes taken into account during classification in different clusters.

Considered databases contain records that characterize people who bought life insurance policies. All the customers, according to their characteristic features were assigned into one of three groups: of high risk—the ones who cause the biggest financial losses for the company; of low risk—the ones who are expected to be rather profitable than harmful; of medium risk—the ones that do not belong neither to first nor to the second group. Finally there are three groups of customers with different level of risk. During experiments, all records from databases were divided into training and test sets. Records of the first ones took part in classification process, while records of the second ones were to check the classification accuracy, by comparing obtained results with real assignments into the groups of risk.

Experiments were divided into three stages. In the first one classification was made on the entire datasets, during the second stage groups of similar customers were found by clustering, and classification was done on each cluster separately. Finally the role of the attributes in obtaining high accuracy is examined.

### A. *Classification on entire data sets*

Results of the classification process, made on entire data sets, were examined taking into account different combinations of attributes. At the beginning, all the attributes were used. The results were rather poor, only 57% of customers were classified correctly in case of the first database (8 attributes)and 59% for the second database (11 attributes) . Much better effects were achieved taking into consideration different combinations of parameters. For the first database, the best results of 71% of objects classified correctly were obtained for combination of 5 from among 8 attributes: “weight”, “smoking”, “profession”, “region”, “disease”. For the second database, also combination of 5 from among 11 attributes: “drinking alcohol”, “smoking”, “disease”, “blood pressure”, “age”; gave the best results of 76% correctly assigned clients.

The main reason for such an improvement in case of usage of selected attributes is the fact that Naïve Bayes algorithm, contrarily to decision trees, cannot recognize which of features have more influence in the classification process. One attribute cannot affect membership of the object in any of the groups. During experiments, it was noticed that including some of the attributes into the classification process can even decrease its efficiency. If number of attributes is not so big, like in considered cases, there may be tried different combinations of attributes to choose the best ones. Another possibility is to ask experts.

### B. *Classification on clusters*

In this part of the experiments, all customers were grouped by unsupervised classification, before applying Naïve Bayes models. To maintain the proper balance between the weight of each customer features, different attributes were taken into account in every stage of the classification. During the

TABLE II
Classification acuracy for all attributes and 3 clusters

| Data set | Correctly classified |
|---|---|
| Cluster 1 | 51.35% |
| Cluster 2 | 65.21% |
| Cluster 3 | 71.5% |
| All data | 63% |

TABLE III
Classification acuracy for all attributes and 2 clusters

| Data set | Correctly classified |
|---|---|
| Cluster 1 | 63.33% |
| Cluster 2 | 75% |
| All data | 68% |

TABLE VI
Classification acuracy for all attributes and 3 clusters

| Data set | Correctly classified |
|---|---|
| Cluster 1 | 79.31% |
| Cluster 2 | 100% |
| Cluster 3 | 65% |
| All data | 74% |

TABLE VII
Classification acuracy for all attributes and 2 clusters

| Data set | Correctly classified |
|---|---|
| Cluster 1 | 73.33% |
| Cluster 2 | 62.5% |
| All data | 69% |

investigations, there were examined the influence of different combinations of attributes as well as the required number of clusters on the accuracy of obtained results. For both data sets, clustering was done according to three customer features, while the other attributes were used to classify clients into appriopriate groups of risk level. There were considered two different approaches: using the same attributes for all the clusters or distinguishing the choice of attributes depending on clusters.

Some exemplary results for the first database (8 attributes, 500 instances), where clustering was done according to the attributes: “drinking alcohol”, “smoking”, “profession”, for different numbers of clusters, are presented in Table II and Table III. In both of the cases the set of 5 other attributes is used for Naïve Bayes classification.

Table IV and Table V shows percentage of correctly nested instances, when classification rules are built by using different attributes for each cluster. The best choice of attributes, measured by the highest accuracy of classification are presented in the last columns.

Comparison of values presented in Table II, Table III, Table IV and Table V shows that the quality of obtained results is better when different attributes are used to build models for each cluster. For example, if the required number of clusters is equal to 3, total accuracy is increased of 17 percentage points. In the case of two clusters the increase was of 10 percentage points. It can be easily noticed that using of two and three clusters gives very similar effects. For three segments, 2 attributes were used for building classification models in each cluster, while for two segments, in both of them, three of five attributes were used. Results presented in the tables were the best from among those obtained in different experiments. In real databases, with greater number of attributes, the opinion of an expert concerning the feature selection may be very useful.

Exemplary results for the second database (11 attributes, 800 instances) are presented in Table VI and Table VII. Similarly to the first database, customers are segmented according to 3 of the following features: “drinking alcohol”, “smoking”, “profession”, with the required number of clusters equal to 2 or 3 and classification models built on the basis of 8 other attributes.

Results for decision rules built on different attributes in each cluster are presented in Table VIII and Table IX. Also in that case, classification effects obtained by using different attributes in each cluster are better. In case of the segmentation into

TABLE IV
Classification acuracy for different attributes and 3 clusters

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 72.97% | weight, disease |
| Cluster 2 | 78.26% | weight, disease |
| Cluster 3 | 87.5% | weight, disease |
| All data | 80% | |

TABLE V
Classification acuracy for different attributes and 2 clusters

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 75% | weight, disease, hobby |
| Cluster 2 | 82.2% | weight, disease, region |
| All data | 78% | |

TABLE VIII
Classification acuracy for different attributes and 3 clusters

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 84.48% | disease, blood pressure, age |
| Cluster 2 | 100% | disease, blood pressure, age |
| Cluster 3 | 77.5% | hobby, disease, age |
| All data | 82% | |

TABLE IX
Classification acuracy for different attributes and 2 clusters

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 81.66% | weight, disease, age |
| Cluster 2 | 77.5% | hobby, disease, age |
| All data | 80% | |

TABLE X
CLASSIFICATION ACURACY FOR ALL ATTRIBUTES AND 2 CLUSTERS

| Data set | Correctly classified |
|---|---|
| Cluster 1 | 55% |
| Cluster 2 | 62.5% |
| All data | 58% |

TABLE XI
CLASSIFICATION ACURACY FOR DIFFERENT ATTRIBUTES AND 2 CLUSTERS

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 66.66% | weight, drinking alcohol, hobby |
| Cluster 2 | 70% | weight, drinking alcohol, hobby |
| All data | 68% | |

two or three clusters, the situation is the same: classification models built for each cluster separately are less complex than the ones obtained for all the data. In all the cases only 3 attributes were used to build models for each cluster. Values of correctly classified instances presented in Table VIII and Table IX, are significantly higher than the ones presented in Table VI and Table VII, where classifications were done on the basis of all 8 attributes.

### C. The choice of attributes

The aim of these investigations, was to check if for all the sets of attributes, segmenting of customers guarantees the improvement of classification results. After several experiments there were indicated attributes: “smoking”, “region”, “disease”, for which results are worse. Table X and Table XI present the effects of classification for the first data set (8 attributes, 500 instances), for clusters built according to those attributes.

It can be easily seen that the results of classification presented in Table X and Table XI are definitely worse than the ones obtained for the entire data set. The accuracy effectively increased after building classification models on different attributes, but it is still worse than the one obtained without using clusters.

## V. EVALUATION

It was shown in the previous section that classification accuracy depends on the choice of attributes as well as the number of clusters. It may be expected that the results of classification should be more accurate for clusters of better quality. Well constructed clusters should have high internal and low external similarity. Calculating of $DB$ index defined by ( 3) allows for choosing the best schema (optimal number of clusters) that will guarantee good quality of obtained clusters. It let us avoid experimental choice of required number of clusters. In considered cases, for the first database Davies Bouldin index is respectively equal to 0.0539 for two clusters and 0.0545 in case of three clusters. Comparisons of Table II and Table III as well as Table IV and Table V show that the effects in both cases are very similar, what is reflected in proximity of $DB$ values. In the case of second database, $DB$ value for two clusters is equal to 0.0547, while for three clusters to 0.0534, which should guarantee the better quality of clusters. Indeed, comparing Table VII and Table VIII as well as Table IX and Table X, it can be easily noticed that the choice of three clusters gives better effects in both cases.

TABLE XII
CLASSIFICATION ACURACY FOR DIFFERENT ATTRIBUTES AND 3 CLUSTERS, AFTER APPLICATION OF THRESHOLD IN EACH CLUSTER

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 72.97% | weight, disease |
| Cluster 2 | 69.56% | weight, disease |
| Cluster 3 | 90% | weight, disease |
| All data | 79% | |

TABLE XIII
CLASSIFICATION ACURACY FOR DIFFERENT ATTRIBUTES AND 2 CLUSTERS AFTER APPLICATION OF TOLERANT TRESHOLD IN EACH CLUSTER

| Data set | Correctly classified | Attributes |
|---|---|---|
| Cluster 1 | 83.33% | weight, disease, age |
| Cluster 2 | 60% | hobby, disease, age |
| All data | 74% | |

The next issue that should be taken into account is a presence of exceptions in data sets. During the segmentation process, such objects, that do not fit into any of the groups, usually are allocated into one of them. Existence of such elements may decrease quality of clusters and efficiency of the classification process. That situation may be avoided by removing the most distant objects from the clusters. The establishing of tolerance thresholds for each cluster may allow to isolate outliers and not to take them into account during classification process. In the current research as the threshold value for each cluster, the maximum distance between two objects in the generation, divided by the number of clusters, is used.

Table XII presents classification results in case of three clusters, for the first database, after application of tolerance threshold. Comparing Table XII and Table IV we can see that accuracy in Cluster 1 did not change, while in Cluster 3, it efficiently increased from 87.5% to 90%. In Cluster 2, however, we can observe a decrease from 78.26 to 69.56%. Similar situation may be noticed in the case of the second database and segmentation into two clusters. The results are presented in Table XIII

Comparison of Table IX and Table XIII shows that efficiency of classification increased in the first cluster but considerably decreased in the second one. It means that different value of tolerance threshold established for different clusters may give better results. If in the case presented in Table XIII we will use tolerance threshold only in the first cluster, we will obtain results with a general increase in accuracy, what can be seen in Table XIV.

TABLE XIV
CLASSIFICATION ACURACY FOR DIFFERENT ATTRIBUTES AND 2 CLUSTERS AFTER APPLICATION OF THRESHOLD IN FIRST CLUSTER

| **Data set** | **Correctly classified** | **Attributes** |
|---|---|---|
| Cluster 1 | 83.33% | weight, disease, age |
| Cluster 2 | 77.5% | hobby, disease, age |
| All data | 81% | |

## VI. CONCLUSION

In the paper, there is considered application of classification techniques for insurance risk evaluation. The idea is based on dividing clients into three groups of a different risk level. There has been chosen Naïve Bayes model as a classifier. Cluster analysis technique is proposed to improve classification accuracy. Experiments conducted on two data sets, characterised by different attributes of different number of records showed that building classification models for each cluster separately, ameliorates the accuracy of obtained results. For the first database number of correctly classified instances increased from 71% in the case of building the same classification model for all the data to 80%, in the case of differentiating classification models according to clusters, while for the second dataset the growth was from 76% to 82%.

The investigations showed that different results may be received for different number of clusters. To avoid determining a number of clusters experimentally, validation technique, which will indicate optimal schema of clusters of the best quality, may be applied. Classification accuracy can be increased by application of restrictions concerning objects in each cluster. However, experiments showed that establishing of a tolerant threshold, may also bring opposite effects. There should be worked out the strategy, that will allow to differentiate threshold values depending on clusters' properties. The experiments proved that classification process may give much better results by combining Naïve Bayes models with cluster analysis. Taking into account the data that insurance companies possess they may easily segment their customers and build risk models for each of the group separately. The experiments were conducted on artificially generated data sets, in the next step obtained results should be verified on the real data of insurance company.

Future research should also consist of development of the proposed model, by using for example SBC instead of Naïve Bayes classification, as the most crucial problem and the big challenge for researchers, at the same time, is the choice of the optimal combination of features for building classification models.

## REFERENCES

[1] H. Zhang, "The optimality of Naïve Bayes,"*in the 17th FLAIRS Conference,* Florida, 2004.

[2] S. A. Klugman, H. H. Panjer and G. E. Willmot, *Loss Models: From Data to Decision,* John Wiley & Sons, New York, 1998.

[3] C. Apte, E. Grossman, E. Pednault, B. Rosen, F. Tipu and B. White, "Probabilistic estimation based data mining for discovering insurance risks,"*IEEE Intelligent Syst.,* vol. 14, 1999, pp. 49–58.

[4] J.-U. Kietz, U. Reimer and M. Staudt, "Mining insurance data," *in the 23rd VLDB Conference,* Athens, Greece, 1997.

[5] M. S. Viveros, J. P. Nearhos and M. J. Rothman, "Applying data mining techniques to a health insurance information system," *in the 22nd International Conference on Very Large Data Bases,* Bombay, India, 1996, pp. 286–294.

[6] S. J. Hong and S. M. Weiss, "Advances in predictive models for data mining," *Pattern Recogn. Lett.,* vol. 22, 2001, pp. 55–61.

[7] R. Mosley, "The use of predictive modeling in the insurance industry," *PINNACLE Actuarial Resources, INC.,* January, 2005.

[8] I. Kolyshkina and R. Brookes,"Data mining approaches to modeling insurance risk," *Report,* PriceWaterhouseCoopers, 2002.

[9] J. Han and M. Kamber, *Data Mining: Concepts and Techniques. Second Edition.* Morgan Kaufmann Publishers, San Francisco CA; 2006.

[10] S. P. D'Arcy, "Predictive modeling in automobile insurance: a preliminary analysis,"*in World Risk and Insurance Economics Congress,* Salt Lake City, Utah, August 2005.

[11] H. He, W. Graco and X. Yao, "Application of genetic algorithms and k-nearest neighbour in medical fraud detection," *In Proceedings of SEAL 1998,* Canberra, Australia, 1999, pp. 74–81.

[12] H. He, J. Wang, W. Graco and S. Hawkins, "Application of neural networks to detection of medical fraud," *Expert Syst. Appl.,* vol. 13, 1997, pp. 329–336.

[13] J. Major and D. Riedinger, "EFD:A hybrid knowledge/statistical -based system for the detection of fraud," *J. Risk Insur.,* vol. 69, 2002, pp. 309–324.

[14] A. F. Shapiro, "The merging neural networks, fuzzy logic and genetic algorithms," *Insur. Math. Econ.,* vol. 31, 2002, pp. 115–131.

[15] A. F. Shapiro, "Fuzzy logic in insurance," *Insur. Math. Econ.,* vol. 35, 2004, pp. 399–424.

[16] R. A. Derrig, K. M. Ostaszewski, "Fuzzy techniques of pattern recognition in risk and claim classification," *J. Risk Insur.,* vol. 62, 1995, pp. 447–482.

[17] S. Viaene, R. A. Derrig, B. Baesens and G. Dedene, "A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection," *J. Risk Insur.,* vol. 69, 2002, pp. 373–421.

[18] S. Viaene, R. A. Derrig and G. Dedene, "A case study of applying boosting Naïve Bayes to claim fraud diagnosis,",*IEEE T. Knowl. Data En.,* vol. 16, 2004, pp. 612–620.

[19] J. Huang, J. Lu, Ch. X. Ling, "Comparing Naïve Bayes, decision trees and SVM with AUC and accuracy," *in Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03),* Melbourne, Florida, USA 2003.

[20] Y. Peng, G. Kou, A. Sabatka, J. Matza, Z. Chen, D. Khazanchi and Y. Shi, "Application of classification methods to individual disability income insurance fraud detection," *in ICCS2007. LNCS 4489,* Y. Shi, G. D. van Albada, J. Dongarra, P. Sloot, Eds., Springer, Berlin Heidelberg, 2007, pp. 852–858.

[21] Ch. Ratanamahatana, D. Gunopulos, "Scaling up the Naïve Bayesian classifier: using decision trees for features selection," *in Proceedings of Workshop on Data Cleaning and Preprocessing (DCAP 2002), at IEEE International Conference on Data Mining (ICDM'02),* Maebashi, Japan, 2002.

[22] P. Langley, S. Sage, "Induction of selective Bayesian classifiers," *in Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence,* Seattle, 1994, pp. 399–406.

[23] S. B. Kotsiantis, P. E. Pintelas, "Increasing the classification accuracy of simple Bayesian classifier," *in AIMSA2004. LNAI 3192,* C. Bussler, D. Fensel, Eds., Springer, Berlin Heidelberg, 2004, pp. 198–207.

[24] H. Zhang, L. Jiang, J. Su, "Hidden Naïve Bayes," *in Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI2005),* AAAIPress, 2004, pp. 919–924.

[25] J. Doughtery, R. Kohavi, M. Shami, "Supervised and unsupervised discretization of continuous features," *in Machine Learning: Proceedings of the Twelth International Conference,* A. Prieditis, S. Russell, Eds., Morgan Kaufmann Publishers, 1995, pp. 194–202.

[26] Y. Freund, R. E. Schapiro, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence,* vol. 14, 1999, pp. 771–780.

[27] Extended Medical Examination,
$http : //www.swisslife.ch/etc/slml/slch/obedl/1/200/$
$316.File.tmp/form_9 15035_g esundheitspruefung.pdf.$

[28] $http : //personalinsure.about.com/od/life/a/aa112805a.htm$

[29] D. Lowd, P. Domingos, “Naive Bayes models for probability estimation,” *in Proceedings of 22nd International Conference on Machine Learning,* Bonn, Germany, 2005.

[30] K. P. Murphy, “Naive Bayes classifiers,” $http : //www.cs.ubc.ca/\ murphyk/Teaching/CS340 - Fall06/$ $reading/NB.pdf.$

[31] S. B. Kotsiantis, “Supervised machine learning: a review of classification,” *Informatica,* vol. 31, 2007, pp. 249–268.

[32] M. N. Murty, P. J. Flynn and A. K. Jain, “Data clustering: a review,” *ACM Comput. Surv.,* vol. 31, 1999, pp. 264–323.

[33] D. Zakrzewska, “On integrating unsupervised and supervised classification for credit risk evaluation,” *Information Technology and Control,* vol. 36, 2007, pp. 98–102.

[34] G. Gan, Cha. Ma and J. Wu, *Data Clustering: Theory, Algorithms and Applications,* ASA-SIAM Series on Statistics and Applied Probability, SIAM: Philadelphia, ASA: Alexandria, 2007.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 145–150

ISBN 978-83-60810-14-9
ISSN 1896-7094

# On a class of defuzzification functionals

Witold Kosiński
Department of Computer Science
Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, 02-008 Warsaw, Poland
Kazimierz Wielki University in Bydgoszcz
Faculty of Mathematics, Physics and Technology
ul. Chodkiewicza 30, 85-064 Bydgoszcz
wkos@pjwstk.edu.pl, wkos@ukw.edu.pl
Telephone: (+48) 22-5844-513
Fax: (+48) 22-4844-501

Wiesław Piasecki
Institute of Computer Science
Department of Electrotechnics and Computer Science
Lublin University of Technology
ul. Nadbystrzycka 36 b, 20-618 Lublin, Poland
faust.faust@poczta.fm
Telephone: (+48) 81 525 20 46
Fax: (+48) 81 538 43 49

***Abstract*—Classical convex fuzzy numbers have many disadvantages. The main one is that every operation on this type of fuzzy numbers induces the growing fuzziness level. Another drawback is that the arithmetic operations defined for them are not complementary, for instance: addition and subtraction. Therefore the first author (W. K.) with his coworkers has proposed the extended model called ordered fuzzy numbers (OFN). The new model overcomes the above mentioned drawbacks and at the same time has the algebra of crisp (non-fuzzy) numbers inside. Ordered fuzzy numbers make possible to utilize the fuzzy arithmetic and to construct the Abelian group of fuzzy numbers and then an algebra. Moreover, in turn out, that four main operations introduced are very suitable for their algorithmisation. The new attitudes demand new defuzzification operators. In the linear case they are described by the well–know functional representation theorem valid in function Banach spaces. The case of nonlinear functionals is more complex, however, it is possible to prove general, uniform approximation formula for nonlinear and continuous functionals in the Banach space of OFN. Counterparts of defuzzification functionals known in the Mamdani approach are also presented, some numerical experimental results are given and conclusions for further research are drawn.**

## I. Introduction

CLASSICAL fuzzy numbers are very special fuzzy sets defined on the universe of all real numbers. Fuzzy numbers are of great importance in fuzzy systems. In the applications, the triangular and the trapezoidal fuzzy numbers are usually used.

There are two commonly accepted methods of dealing with fuzzy numbers, both basing on the classical concept of fuzzy sets, namely on the membership functions. The first, more general approach deals with the so-called convex fuzzy numbers of Nguyen [31], while the second one deals with shape functions and $L-R$ numbers, set up by Dubois and Prade [6].

When operating on convex fuzzy numbers we have the interval arithmetic for our disposal. However, the approximations of shape functions and operations are needed, if one wants to remain within the $L-R$ numbers while following the Zadehï¿¡s extension principle [38]. In this representation (in most cases) calculation results are not exact and are questionable if some rigorous and exact data are needed, e.g. in the control or modeling problems. This can be treated as a drawback of the properties of classical fuzzy algebraic operations.

In the literature, moreover, it is well know that unexpected and uncontrollable results of repeatedly applied operations, caused by the need of making intermediate approximations (remarked in [34],[35]) can appear. This rises the heavy argument for those who still criticize the fuzzy number calculus. Fortunately, it was already noticed by both Dubois and Prade in their recent publication [8] that something is missing in the definition of the fuzzy numbers and the operations on them.

In most cases one assumes that a typical membership function ïż¡$\mu_A$ of a fuzzy number $A$ satisfies convexity assumptions requiring after Nguyen [31] all $\alpha$-cuts and the support of $A$ to be convex subsets of $\mathbf{R}$. At this stage it seems necessary to recall both notions used: the $\alpha$-cut of A is a (classical) set $A[\alpha] = \{x \in \mathbf{R} : \mu_A(x) \geq \alpha\}$, for each $\alpha \in [0,1]\}$, and the support of $A$ is the (classical) set $\text{supp}(A) = \{x \in \mathbf{R} : \ \mu_A(x) > 0\}$. One additionally assumes [2], [3], [5], [12], [31], [34] that the convex fuzzy number $A$ has its core, i.e. the (classical) set of those $x \in \mathbf{R}$ for which its membership function $\mu_A(x) = 1$, which is not empty and its support is bounded. Then the arithmetic of fuzzy numbers can be developed using both the Zadeh's extension principle [38], [39] and the $\alpha$-cut with interval arithmetic method [12].

As long as one works with fuzzy numbers that possess continuous membership functions, the two procedures: the extension principle and the $\alpha - cut$ and interval arithmetic method give the same results (cf. [2]). The results of multiple operations on convex fuzzy numbers are leading to a large growth of the fuzziness, and depend on the order of the operations since the distributive law, which involves the interaction of addition and multiplication, does hold there. Moreover, the use of the extension principle in the definition of the arithmetic operations on fuzzy numbers is generally numerically inefficient. These operations cannot be equipped with a linear structure and hence no norm can be defined on them. Standard algebraic operations on fuzzy numbers

978-83-60810-14-9/08/$25.00 © 2008 IEEE

basing on the Zadehïż¡s extension principle and those for fuzzy numbers of $L - R$ type or convex fuzzy numbers (see [5]) which are making use of interval analysis have several drawbacks. They are listed in our previous publications [22],[23], [24],[26], [16].

In our opinion the main drawback is the lack of a solution $X$ to the most simple fuzzy arithmetic equation

$$A + X = C \tag{1}$$

with known fuzzy numbers $A$ and $C$. If the support of $C$ is greater than that of $A$ a unique solution in the form of a fuzzy number $X$ exists. However, this is the only case, since for $A$ with larger support than that of C the solution does not exist. Another drawback is related to the fact that in general $A + B - A$ is not equal to $B$.

The goal of the authors of the previous papers [22], [23], [24], [25], [27] was to overcome the above mentioned drawbacks by constructing a revised concept of fuzzy number and at the same time to have the algebra of crisp (non-fuzzy) numbers inside the concept.

In our investigations we wanted to omit, to some extend, the arithmetic based on the $\alpha$–cut of membership functions of fuzzy numbers (sets), and to be close to the operations known from the real line. It was noticed by Dubois and Prade [7] in 2005, (and repeated recently in [9]) after our definition of the *ordered fuzzy number* [26] had been given, that the concept of fuzzy number is too close to the concept of interval. Our new concept makes it possible to utilize the fuzzy arithmetic in a simple way and to construct an Abelian group of fuzzy numbers, and then an algebra. At the same time the new model contains the cone of convex fuzzy numbers. The definition presented here contains all continuous convex fuzzy numbers, however, its recent enlargement presented in [19] includes all convex fuzzy numbers. Moreover, the new model contains more elements and each convex fuzzy number leads to two different new fuzzy numbers, called here the *ordered fuzzy numbers*, which differ by their orientation. This will become more evident later. Additionally, in turns out that the four main operations introduced are very suitable for algorithmisation. We should stress, however, that the arithmetic of the new model restricted to convex (continuous) fuzzy numbers gives different results in comparison to that of the interval arithmetic. This is evident already in the scalar multiplication and subtraction. However, this gives us the chance to solve the arithmetic equation (1) for any pair of fuzzy numbers $A$ and $C$.

The organization of the paper is as follows. In Section 2 we repeat our main definition and basic properties of extended model of fuzzy numbers presented in the series of papers [15], [16], [21], [22], [23], [24], [25], [26]. Then defuzzification functionals are discussed. First, the linear case, then the nonlinear one. Then a counterpart of the Mamdani center of gravity defuzzification functional is derived. In the final section conclusions together with numerical results of some experiments with implementations of the derived formula are presented.

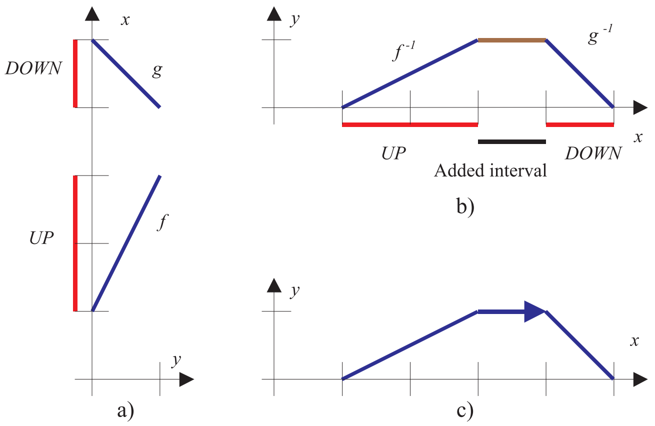


Fig. 1. a) Example of an ordered fuzzy number; b) construction of the membership function; c) the arrow denotes the orientation and the order of inverted functions: first $f$ and then $g$.

## II. Basic Properties of Ordered Fuzzy Numbers

**Definition 1.** By an *ordered fuzzy number* $A$ we mean an ordered pair $(f, g)$ of functions such that $f, g : [0, 1] \to \mathbf{R}$ are continuous.

Notice that in our definition we do not require that two continuous functions $f$ and $g$ are (partial) inverses of some membership function. Moreover, it may happen that membership function corresponding to $A$ does not exist. We call the corresponding elements: $f$—the up-part and $g$—the down-part of the fuzzy number $A$. To be in agreement with further and classical denotations of fuzzy sets (numbers), the independent variable of the both functions $f$ and $g$ is denoted by $y$, and their values by $x$. The continuity of both parts implies their images are bounded intervals, say $UP$ and $DOWN$, respectively (Fig. 1). We have used symbols to mark boundaries for $UP = [l_A, 1_A^-]$ and for $DOWN = [1_A^+, p_A]$. In general, the functions $f$ and $g$ need not to be invertible as functions of $y$, only continuity is required. If we assume, however, that 1) they are monotonous: $f$ is increasing, and $g$ is decreasing, and such that 2) $f \leq g$ (pointwise), we may define the membership function $\mu(x) = f^{-1}(x)$, if $x \in [f(0), f(1)] = [l_A, 1_A^-]$, and $\mu(x) = g^{-1}(x)$, if $x \in [g(1), g(0)] = [1_A^+, p_A]$ and ïż¡$\mu(x) = 1$ when $x \in [1_A^-, 1_A^+]$. In this way we obtain the membership function ïż¡$\mu(x), x \in \mathbf{R}$. When the functions $f$ and/or $g$ are not invertible or the condition 2) is not satisfied then in the plane $x - y$ the membership curve (or relation) can be defined, composed of the graphs of $f$ and $g$ and the line $y = 1$ over the core $\{x \in [f(1), g(1)]\}$.

Notice that in general $f(1)$ needs not be less than $g(1)$ which means that we can reach improper intervals, which have been already discussed in the framework of the extended interval arithmetic by Kaucher [11]. In such case Prokopowicz has introduced in [33] the *corresponding* membership function which can be defined by the formulae:

$$\mu(x) = \max \arg\{f(y) = x, g(y) = x\} \tag{2}$$
$$\text{if } x \in \text{Range}(f) \cup \text{Range}(g) \ ,$$

$$\mu(x) = 1 \quad \text{if } x \in [f(1), g(1)] \cup [g(1), f(1)], \tag{3}$$
$$\text{and } \mu(x) = 0, \text{ otherwise}, \tag{4}$$

where one of the intervals $[f(1), g(1)]$ or $[g(1), f(1)]$ may be empty, depending on the sign of $f(1) - g(1)$, (i.e., if the sign is $-1$ then the second interval is empty).

### A. Norm and partial order

Let $\mathcal{R}$ be a universe of all OFN's. Notice that this set is composed of all pairs of continuous functions defined on the closed interval $I = [0, 1]$, and is isomorphic to the linear space of real 2D-vector valued functions defined on the unit interval $I$ with the norm of $\mathcal{R}$ as follows

$$||A|| = \max(\sup_{s\in I} |f_A(s)|, \sup_{s\in I} |g_A(s)|) \text{ if } A = (f_A, g_A) \ .$$

The space $\mathcal{R}$ is topologically a Banach space. The neutral element of addition in $\mathcal{R}$ is a pair of constant functions equal to crisp zero. It is also a Banach algebra with unity: the multiplication has a neutral element – the pair of two constant functions equal to one, i.e., the crisp one.

A relation of partial ordering in $\mathcal{R}$ can be introduced by defining the subset of 'positive' ordered fuzzy numbers: a number $A = (f, g)$ is not less than zero, and by writing

$$A \geq 0 \quad \text{iff} \quad f \geq 0, g \geq 0 \ . \tag{5}$$

In this way the set $\mathcal{R}$ becomes a partially ordered ring.

## III. Representation of Defuzzification Functional

Defuzzification is a main operation in fuzzy controllers and fuzzy inference systems where fuzzy inference rules appear, in the course of which to a membership function representing classical fuzzy set a real number is attached. We know a number of defuzzification procedures from the literature. Since classical fuzzy numbers are particular case of fuzzy sets the same problem appears when rule's consequent part is a fuzzy number. Then the problem arises what can be done when a generalization of classical fuzzy number in the form of an ordered fuzzy number follows? Are the same defuzzification procedures applicable? The answer is partial positive: if the ordered fuzzy number is *proper* one, i.e. its membership relation is a function, then the same procedure can be applied. What to do, however, when the number is improper, i.e. the membership relation is by no means of functional type?

In the case of fuzzy rules in which ordered fuzzy numbers appear as their consequent part we need to introduce a new defuzzification procedure. In this case the concept of functional, even linear, which maps elements of the Banach space into reals, will be useful.

The Banach space $\mathcal{R}$ with its Tichonov product topology of $C([0, 1]) \times C([0, 1])$, with $C([0, 1])$ the Banach space of continuous functions on $[0, 1]$, may lead to a general representation of linear and continuous functional on $\mathcal{R}$. According to the Banach-Kakutami-Riesz representation theorem any linear and continuous functional $\bar{\phi}$ on the Banach space $C([0, 1])$ is uniquely determined by a Radon measure $\nu$ on $S$ such that

$$\bar{\phi}(f) = \int_{[0,1]} f(s)\nu(ds) \quad \text{where } f \in C([0, 1]) \ . \tag{6}$$

It is useful to remind that a Radon measure is a regular signed Borel measure (or differently: a difference of two positive Borel measures). A Boreal measure is a measure defined on a $\sigma$-additive family of subsets of $[0, 1]$ which contains all open subsets.

However, on the interval $[0, 1]$ each Radon measure is represented by a Stieltjes integral [29] with respect to a function of a bounded variation. Hence we can say that for any continuous functional $\bar{\phi}$ on $C([0, 1])$ there is a function of bounded variation $h_\phi$ such that

$$\bar{\phi}(f) = \int_0^1 f(s) dh_\phi(s) \quad \text{where } f \in C([0, 1]) \ . \tag{7}$$

Hence we may say that due to the representations (6) and (7) *any linear and bounded functional $\phi$ on the space $\mathcal{R}$ can be identified with a pair of functions of bounded variation through the following relationship*

$$\phi(f, g) = \int_0^1 f(s) dh_1(s) + \int_0^1 g(s) dh_2(s) \tag{8}$$

where the pair of continuous functions $(f, g) \in \mathcal{R}$ represents an ordered fuzzy number and $h_1, h_2$ are two functions of bounded variation on $[0, 1]$.

From the above formula an infinite number of defuzzification procedures can be defined. The standard defuzzification procedure in terms of the area under the membership relation can be defined; it is realized by a linear combinations of two Lebesgue measures of $[0, 1]$. In the present case, however, the area is calculated in the $y$-variable, since the ordered fuzzy number is represented by a pair of continuous functions in the $y$ variable (cf. (2)). Moreover, to each point $s \in [0, 1]$ a Dirac delta (an atom) measure can be related, and such a measure represents a linear and bounded functional which realizes the corresponding defuzzification procedure. For such a functional, a sum (or in a more general case – a linear combination $af(s)+bg(s)$) of their values is attached to a pair of functions $(f, g)$ at this point.

For example, if we take the Dirac atomic measure concentrated at $s = 1$, and define

$$\nu_1 = a\delta_1 \text{ i } \nu_2 = b\delta_1$$

where $\delta_1$ is the atomic measure of $\{1\}$, then the value of the defuzzification operator (functional) in (8), denoted here by $\phi_m$ and calculated at $A = (f_A, g_A)$ will be

$$\phi_m(A) = af_A(1) + bg_A(1) \tag{9}$$

and if $a + b = 1/2$, then it is a mean value of both functions (from the core of $f_A$ and $g_A$).

A different choice of the measures may lead to the surface area under the graph of the function, and the first moment of inertia. For example, if

$$\nu_1 = a(s)\lambda \text{ and } \nu_2 = b(s)\lambda$$

where $\lambda$ is the Lebesgue measure of the interval $[0, 1]$ of the real line, and $a(s), b(s)$ are integrable functions on the interval, then in the case of a positive oriented number $A = (f_A, g_A)$ with $f_A \leq g_A$ and

$$b(s) = -a(s) = 1 \tag{10}$$

the defuzzification functional (8) calculated at $A = (f_A, g_A)$ will give the surface area contained between graphs of $f_A$ and $g_A$. If, however, in (10) we put

$$b(s) = -a(s) = s \tag{11}$$

we will get the first moment of inertia of this area.

## IV. Nonlinear Defuzzification Functionals

It is evident that nonlinear and multivariant function compositions of linear functionals will lead to nonlinear defuzzification functionals. For example, a ratio being a nonlinear composition of two linear functionals, where the first one is the first moment and the second . the surface area, discussed in the previous subsection, may lead to the center of gravity known from the Mamdani approach, however, with respect to $s = y$ variable.

Now we can state a uniform approximation theorem concerning the defuzzification operators (functionals). To this end let us use the following denotation. Let $\mathcal{A} \subset \mathcal{R}$ be a subset of all ordered fuzzy numbers $\mathcal{R}$ formed of pairs of functions which are equi-continuous and equi-bounded. Notice that from the theorem of Ascoli-Arzelà [36] it follows that a subset of $C([0, 1])$ is compact if its elements are equi-continuous and equi-bounded. By $\mathcal{G}$ we denote the set of all multivariant continuous functions defined on the appropriate Cartesian product of the set of real numbers. In other words $F \in \mathcal{G}$ if there is a natural number $k$ such that $F : \mathbf{R}^k \to \mathbf{R}$ and $F$ is continuous in the natural norm of $\mathbf{R}^k$. By $\mathcal{D}$ we denote the set of all linear and continuous functionals defined on $\mathcal{A} \subset \mathcal{R}$ (compare (8)). Here we could identify the set $\mathcal{D}$ with the adjoint space $\mathcal{R}^*$ since each continuous (bounded) and linear functional on the whole space $\mathcal{R}$ is a also continuous, linear functional on each subspace, hence on the subset $\mathcal{A}$. Moreover, each continuous, linear functional on a subspace $\mathcal{A} \subset \mathcal{R}$ can be extended to the whole space $\mathcal{R}$, thanks to the Hahn-Banach theorem [36]. Let us use the denotation $\mathcal{R}^\diamond$ for the space of all continuous (not necessarily linear) functionals from $\mathcal{R}$ into reals $\mathbf{R}$. Notice that $\mathcal{R}^\diamond \supset \mathcal{R}^*$.

If a function $F$ of $k$ variables is from $\mathcal{G}$ and $\varphi_1, \varphi_2, ..., \varphi_k \in \mathcal{D}$ then their superposition $F \circ (\varphi_1, \varphi_2, ..., \varphi_k)$ is a function from $\mathcal{D}$ into $\mathbf{R}$, i.e., the functional

$$F \circ (\varphi_1, \varphi_2, ..., \varphi_k) : \mathcal{D} \to \mathbf{R}, \text{ with } F \in \mathcal{G},\ \varphi_1, \varphi_2, ..., \varphi_k \in \mathcal{D} \tag{12}$$

is a defuzzification operator, nonlinear in general. To make the notation short we will write

$$F \circ (\varphi_1, \varphi_2, ..., \varphi_k) =: F(\varphi_1, \varphi_2, ..., \varphi_k). \tag{13}$$

**Theorem.** *Let $\mathcal{A} \subset \mathcal{R}$ be a compact subset of the space of all ordered fuzzy numbers $\mathcal{R}$, and let $\mathcal{D}$ be the set of all linear and continuous functionals defined on $\mathcal{A}$, and let $\mathcal{G}$ be the set of all multivariant continuous functions defined on the appropriate Cartesian product of the set of real numbers. Then the set $\mathcal{H}$ composed of all possible compositions (superpositions) of the type (12) where $F$ is from $\mathcal{G}$ and $\varphi_1, \varphi_2, ..., \varphi_k$ are from $\mathcal{D}$, with arbitrary $k$, is dense in the space $\mathcal{R}^\diamond$ of all continuous functionals from $\mathcal{R}$ into reals* ***R****.*
The proof based on the classical Stone–Weierstrass theorem will be published in another paper.

## V. Center of Gravity

Let us stay with the case of nonlinear functionals. It will be a functional corresponding to that know for convex fuzzy numbers and representing the center of gravity of the area under the graph of the membership function. In the case of OFN the membership function does not exist, in general, however, we may follow to some extend the previous construction.

Let consider an ordered fuzzy number $A = (f, g)$ given in Fig.4. Since $f(g) > g(s)$ for any $s \in [0, 1]$ then by adding an interval (perpendicular to the $s$-axis) which joints the points $(1, f(1))$ and $(1, g(1))$ we get a figure (an area) bounded by the graphs of $f(s)$ and $g(s)$, and the $t$–axis. Our aim is to determine the $t$–th coordinate of the center of gravity of this figure.

Assuming, as it is natural, that the density of each point of the figure is the same and equal to one, first we calculate the moment of inertia of this figure with respect to the $t$–axis. Here $t$ denotes $x$ variable. Let us consider a differential (incremental) element $ds$ situated between the coordinate values $s_1$ and $s_2$ and the corresponding piece of the figure above. Its moment $M_s$ is the product of the area and the length of the arm with respect the $t$–axis, which is the local center of gravity. The (incremental) area is equal to $[f(s_1) - g(s_1)]ds$, while the center of gravity of this area (its $t$–coordinate) can be approximated by the middle point of the interval of $[g(s_1), f(s_1)]$ as $\dfrac{f(s_1) - g(s_1)}{2} + g(s_1) = \dfrac{f(s_1) + g(s_1)}{2}$.

Hence we have for the moment of inertia of this differential (incremental) area element the expression

$$\frac{f(s_1) + g(s_1)}{2}[f(s_1) - g(s_1)]ds\,.$$

Since the point $s_1$ has been chosen quite arbitrarily, the moment of inertia of the whole figure bounded by the graphs of the functions $f$ and $g$, $t$–axis and the interval bounding the points $(1, f(1))$ and $(1, g(1))$, will be the integral

$$M = \int_0^1 \frac{f(s) + g(s)}{2}[f(s) - g(s)]ds \tag{14}$$

Let $P$ be the mass of the figure equal to the area of the whole figure, due to our assumption about the homogeneous distribution of the mass,

$$P = \int_0^1 [f(s) - g(s)]ds\,. \qquad (15)$$

Now we use the classical balance equation of inertial momentum which states that the moment of inertia $M$ is equal to the product of the mass of the figure $P$ and the arm $r$ (of the figure, which is the $t$ coordinate of the global center of gravity),

$$M = P \cdot r\,. \qquad (16)$$

The coordinate $r$ is wanted value of the defuzzification functional representing the center of gravity. Having the expressions (14) and (15) we end up with the following expression for the center of gravity defuzzification functional $\phi_G$ calculated at OFN $(f, g)$

$$\phi_G(f,g) = \int_0^1 \frac{f(s)+g(s)}{2}[f(s)-g(s)]ds\{\int_0^1 [f(s)-g(s)]ds\}^{-1}\,. \qquad (17)$$

We can see that the functional $\phi_G$, denoted on the figures below by COG, is nonlinear. Our derivation is based on the Eq. (16) and the assumption that the function $f(s) \geq g(s)$ which may not be fulfilled in any case. However, if the opposite inequality holds the value does not change. The case when the none of them is true is more complex, we are going, however, to adapt this representation for any ordered fuzzy number $(f, g)$.

## VI. Conclusions

The ordered fuzzy numbers are tool for describing and processing vague information. They expand existing ideas. Their "good"ïż¡ algebra opens new areas for calculations. Beside that, new property – *orientation* can open new areas for using fuzzy numbers. Important fact (in authors' opinion) is that thanks to OFNs we can supply without complication the classical field of fuzzy numbers with new ideas. We can use the OFNs instead of convex fuzzy numbers, and if we need to use extended properties we can use them easily. One of directions of the future work with the OFNs is the construction of new class of nonlinear defuzzification functionals based on required properties. The above problem may have several solutions; however, one can look for one of them with the help of Theorem . Since the Weierstrass theorem states that each continuous function (of many variables) defined on a compact set can be approximated with a given accuracy by a polynomial (of many variables) of an appropriate, i.e., sufficiently high order, then with the use of our final result the family $\mathcal{H}$ may be taken as a set of polynomials of many variables. In the recent papers [18], [20] propositions concerning specially dedicated evolutionary algorithms for the determination of the approximate form of the functional have been discussed. Some interpretations and applications of the present approach to fuzzy modeling, control and finance have been presented in the recent publications [17],[19].

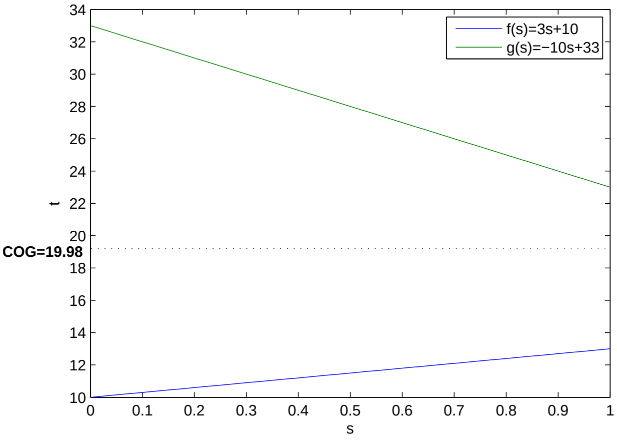


Fig. 2. Ordered fuzzy numbers with affine functions $f, g$ and its center of gravity.

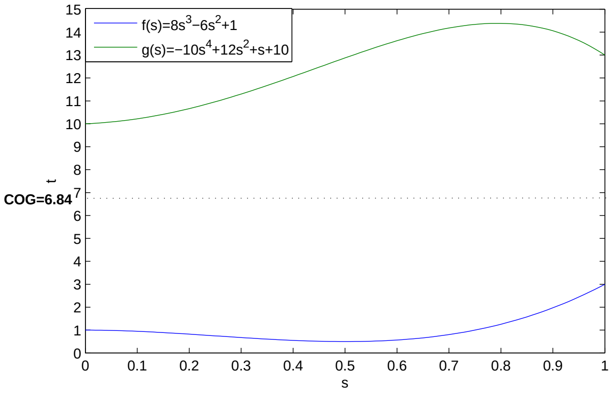


Fig. 3. Ordered fuzzy numbers with polynomial functions $f, g$ and its center of gravity.

Numerical results of implementations of the derived formula for the center of gravity functional $\phi_G$ =:COG from Eq. (17) for two ordered fuzzy numbers, with the functions $f$ and $g$ of the affine type (Fig. 2) and of the polynomial type (Fig. 3), respectively, are presented below. On the figures the variable $t$ corresponds to $x$, and the variable $s$ to $y$, in Fig.1.

## References

[1] Buckley James J. (1992), Solving fuzzy equations in economics and finance, *Fuzzy Sets and Systems*, **48**, 289–296.

[2] Buckley James J. and Eslami E. (2005), *An Introduction to Fuzzy Logic and Fuzzy Sets*, Physica-Verlag, A Springer-Verlag Company, Heidelberg.

[3] Chen Guanrong, Pham Trung Tat, (2001), *Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*, CRS Press, Boca Raton, London, New York, Washington, D.C.

[4] Czogała E., Pedrycz W. (1985), *Elements and Methods of Fuzzy Set Theory* (in Polish), PWN, Warszawa, Poland.

[5] Drewniak J.(2001), Fuzzy numbers (In Polish), in: *Fuzzy Sets and their Applications*(In Polish),J. Chojcan, J. Łęski (Eds.), WPŚ, Gliwice, Poland, pp. 103–129.

[6] Dubois D., H. Prade H. (1978), Operations on fuzzy numbers, *Int. J. System Science*, **9** (6), 613–626.

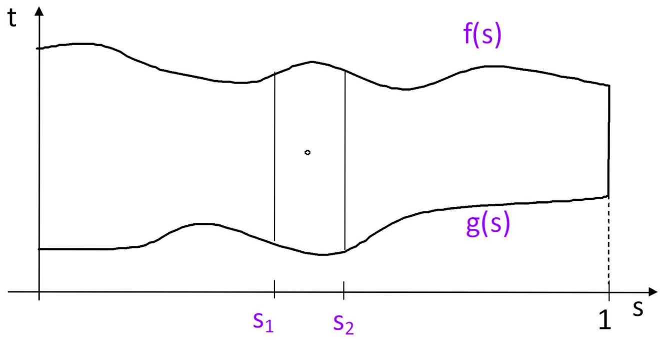


Fig. 4. Sketch of ordered fuzzy number $(f, g)$ and the corresponding area.

[7] Dubois D., Fargier. H, Fortin J.(2005), A generalized vertex method for computing with fuzzy intervals, in *Proc. IEEE Int. Conf. Fuzzy Syst., Budapest, Hungary, 2004*, pp. 541–546.

[8] Dubois D., H. Prade H. (2008), Gradual elements in a fuzzy set, Soft. Comput., **12**, 165–175, DOI 10.1007/s00500-007-0187-6.

[9] Fortin J., Dubois D., Fargier H. (2008), Gradual numbers and their application to fuzzy interval analysis, *IEEE Trans. Fuzzy Syst.*, **16**(2), 388–402, DOI 10.1109/TFUZZ.2006.890680.

[10] Goetschel R. Jr., Voxman W. (1986), Elementary fuzzy calculus, *Fuzzy Sets and Systems*, **18** (1), 31–43.

[11] Kaucher E. (1980), Interval analysis in the extended interval space IR, *Computing, Suppl.* **2**, 33–49.

[12] Kaufman A. and Gupta M. M. (1991), *Introduction to Fuzzy Arithmetic*, Van Nostrand Reinhold, New York.

[13] Kacprzyk J. (1986), *Fuzzy Sets in System Analysis* (in Polish) PWN, Warszawa, Poland.

[14] Klir G.J. (1997), Fuzzy arithmetic with requisite constraints, *Fuzzy Sets and Systems*, **91**(2), 165–175.

[15] Koleśnik R., Prokopowicz P., Kosiński W. (2004), Fuzzy Calculator – usefull tool for programming with fuzzy algebra, in *Artficial Intelligence and Soft Computing – ICAISC 2004, 7th Int. Conference, Zakopane, Poland, June 2004*, L. Rutkowski, Jörg Siekmann, Ryszard Tadeusiewicz, Lofti A. Zadeh (Eds.) Lecture Notes on Artificial Intelligence, vol. 3070, pp. pp. 320–325, Springer-Verlag, Berlin, Heidelberg, 2004.

[16] Kosiński W. (2004), On defuzzyfication of ordered fuzzy numbers, in: *ICAISC 2004, 7th Int. Conference, Zakopane, Poland, June 2004*, L. Rutkowski, Jörg Siekmann, Ryszard Tadeusiewicz, Lofti A. Zadeh (Eds.) LNAI, vol. 3070, pp. 326–331, Springer-Verlag, Berlin, Heidelberg, 2004.

[17] Kosiński W., (2006), On soft computing and modelling, *Image Processing Communications*, An International Journal with special section: Technologies of Data Transmission and Processing, held in 5th International Conference INTERPOR 2006, **11**, (1), 71–82.

[18] Kosiński W., Markowska-Kaczmar Urszula (2007), An evolutionary algorithm determining a defuzzyfication functional, *Task Quarterly*, **11**, (1-2) 47–58.

[19] Kosiński W. (2006), On fuzzy number calculus, *Int. J. Appl. Math. Comput. Sci.*, **16** (1), 51–57.

[20] Kosiński W. (2007), Evolutionary algorithm determining defuzzyfication operators, *Engineering Applications of Artificial Intelligence*,ïż¡**20** (5),ïż¡ 619–627 ,DOI 10.1016/j.engappai.2007.03.003

[21] Kosiński W., Prokopowicz P. (2004), Algebra of fuzzy numbers (In Polish: Algebra liczb rozmytych), *Matematyka Stosowana. Matematyka dla Społeczeństwa*, **5** (46), 37–63.

[22] Kosiński W., Piechór K., Prokopowicz P. (2001), Tyburek K.: On algorithmic approach to operations on fuzzy numbers, in: *Methods of Artificial Intelligence in Mechanics and Mechanical Engineering*, T. Burczyński, W. Cholewa (Eds.), pp. 95–98, PACM, Gliwice, Poland.

[23] Kosiński W., Prokopowicz P. , Ślęzak D. (2002), Fuzzy numbers with algebraic operations: algorithmic approach, in: *Intelligent Information Systems 2002*, M. Klopotek, S. T. Wierzchoń, M. Michalewicz(Eds.) Proc.IIS'2002, Sopot, June 3–6, 2002, Poland, pp. 311–320, Physica Verlag, Heidelberg, 2002.

[24] Kosiński W., Prokopowicz P., Ślęzak D. (2002), Drawback of fuzzy arithmetics—new intutions and propositions, in: *Proc. Methods of Aritificial Intelligence*, T. Burczyński, W. Cholewa, W. Moczulski(Eds.), pp. 231–237, PACM, Gliwice, Poland, 2002.

[25] Kosiński W., P. Prokopowicz P., Ślęzak D. (2003), On algebraic operations on fuzzy numbers, in *Intelligent Information Processing and Web Mining*, Proc. of the International IIS: IIPWM,03 Conference held in Zakopane, Poland, June 2-5,2003, M. Klopotek, S. T. Wierzchoń, K. Trojanowski(Eds.), pp. 353–362, Physica Verlag, Heidelberg, 2003.

[26] Kosiński W., Prokopowicz P., Ślęzak D.(2003), Ordered fuzzy numbers, *Bulletin of the Polish Academy of Sciences, Sér. Sci. Math.*, **51** (3), 327–338.

[27] Kosiński W., Słysz P.(1993), Fuzzy numbers and their quotient space with algebraic operations, *Bull. Polish Acad. Scien.*, **41/3** 285—295.

[28] Kosiński W., Weigl M.(1998), General mapping approximation problems solving by neural networks and fuzzy inference systems, *Systems Analysis Modelling Simulation*, **30** (1), 11–28.

[29] Łojasiewicz S. (1973), *Introduction to the Theory of Real Functions*, (In Polish: Wstęp do teorii funkcji rzeczywistych), Biblioteka Matematyczna, Tom 46, PWN, Warszawa, 1973.

[30] Martos B. (1983), *Nonlinear Programming – Theory and Methods*, PWN, Warszawa, Poland (Polish translation of the English original published by Akadémiai Kiadó, Budapest, 1975).

[31] Nguyen H.T. (1978), A note on the extension principle for fuzzy sets, *J. Math. Anal. Appl.* **64**, 369–380.

[32] Piegat A.(1999), *Fuzzy Modelling and Control*, (In Polish: Modelowanie i sterowanie rozmyte), Akademicka Oficyna Wydawnicza PLJ, Warszawa.

[33] Prokopowicz P. (2005), *Algorithmization of Operations on Fuzzy Numbers and its Applications* (In Polish: Algorytmizacja działań na liczbach rozmytych i jej zastosowania), Ph. D. Thesis, IPPT PAN, kwiecień 2005.

[34] Wagenknecht M. (2001), On the approximate treatment of fuzzy arithmetics by inclusion, linear regression and information content estimation, in: *Fuzzy Sets and their Applications* (In Polish), J. Chojcan, J. Łęski (eds.), Wydawnictwo Politechniki Śląskiej, Gliwice, 291–310.

[35] Wagenknecht M., Hampel R., Schneider V.(2001), Computational aspects of fuzzy arithmetic based on archimedean $t$-norms, *Fuzzy Sets and Systems*, **123** (1), 49–62.

[36] Yoshida K. (1980), *Functional Analysis*, Sixth edition, Springer-Verlag, Berlin, Heidelberg, New York, 1980.

[37] Zadeh L. A.(1965), Fuzzy sets, *Information and Control*, **8** (3), 338–353.

[38] Zadeh L. A. (1975), The concept of a linguistic variable and its application to approximate reasoning, Part I, *Information Sciences*, **8**(3), 199–249.

[39] Zadeh L. A.(1983), The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems*, **11**(3), 199–227.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Assessing the Properties of the World Health Organization's Quality of Life Index

Tamar Kakiashvili, Waldemar W. Koczkodaj*,
Phyllis Montgomery, Kalpdrum Passi
Laurentian University, Sudbury, Ontario, Canada
*the corresponding author: wkoczkodaj@cs.laurentian.ca

Ryszard Tadeusiewicz
AGH University of Science and Technology, Poland

***Abstract*—This methodological study demonstrates how to strengthen the commonly used World Health Organization's Quality of Life Index (WHOQOL) by using the consistency-driven pairwise comparisons (CDPC) method. From a conceptual view, there is little doubt that all 26 items have exactly equal importance or contribution to assessing quality of life. Computing new weights for all individual items, however, would be a step forward since it seems reasonable to assume that all individual questions have equal contribution to the measure of quality of life. The findings indicate that incorporating differences of importance of individual questions into the model is essential enhancement of the instrument.**

***Index Terms*—Quality of life, inconsistency analysis, consistency-driven pairwise comparisons**

## I. Introduction

DURING the past three decades, many different quality of life measures have been developed for use as an indicator of patient-focused health outcomes. Commonly used quality of life measures had reported psychometric properties. One measure in particular, the World Health Organization Quality of Life (WHOQOL) measure has been widely field tested since its inception in 1991. This measure was developed to an international cross-culturally comparable quality of life assessment for clinical populations. Its purpose is to assess subjects' perceptions of their quality of life in the context of their personal values and beliefs as well as their social culture. There are three versions of this measure. The first contains 100-items and is commonly used in large clinical trails. The brief version, WHOQOL-BREF, consists of 26-items.

A third version available on the web (www.who.int/mental_health/media/en/76.pdf) seems to be a modification of what was previously published in [9].

To supplement to the WHOQOL User's manual published in 1998, this research will further examine the psychometric properties of the measure by using consistency-driven pairwise comparison. The assumption of this approach is that not all instrument items are of equal importance and including the relative importance in the model contribute to the enhancement of the measure.

## II. The Pairwise Comparison Preliminaries and a Quality of Life Model

From a mathematical point of view, the pairwise comparisons method [2],[3],[4], Appendix A creates a matrix (for example, A) of values $(aij)$ of the $i$-th candidate (or alternative) compared head-to-head (one-on-one) with the $j$-th candidate. A scale $[1/c, c]$ is used for $i$ to $j$ comparisons where $c > 1$ is a not-too-large real number (5 to 9 in most practical applications).

It is usually assumed that all the values $(aij)$ on the main diagonal are 1 (the case of $i$ compared with $i$ and that $A$ is reciprocal: $(aij) = 1/(aij)$ since $i$ to $j$ is (or at least, is expected to be) the reciprocal of $j$ to $i$. (As explained below, the reciprocity condition is not automatic in certain scenarios of comparisons.) It is fair to assume that we are powerless, or almost powerless, as far as inconsistency is concerned. All we can do is to locate it and reconsider our own comparisons to reduce the inconsistency in the next round.

An pioneer of this method is Condorcet [5]. He used the pairwise comparisons in 1785 in the context of counting political ballots. In 1860, however, Fechner provided further yet limited psychometric information about this method. By way of refining the method, Thurstone [7] described pairwise comparisons method as a statistical analysis and proposed a solution. In 1977, Saaty [8] introduced a hierarchy instrumental for practical applications.

The WHOQOL addresses four domains: physical health (PH), psychological health (PSYCH), social relationships (SR), and environment (ENV). Using the Saaty's heuristic approach of having no more than seven items in one group, the ENV domain was mechanically subdivide into ENV1 and ENV2 since it has eight objects. Implementing the Concluder system, the results are shown in Fig. 1. This figure is an illustration of the full model due to screen limitation and scrolling. The items listed on the left-hand side of Fig. 1 should be attached to the first level (groups) to create a hierarchical structure.

Using a scale 1 to 5, the relative importance of each of the five groups were entered and compared objects in the smallest subgroup. For example, SR (social relationships) was compared against each other. For example, "personal relationship" and "sex" are compared to each other in the subgroup SR (social relationship) and given 4 out of 5 (which can be changed for every clinical case to which this instrument is applied). The results are presented in Table 1.

Clearly, the above matrix was not consistent since $a_{13} = 4$. It did not equal to $a_{12} * a_{23}$. To address the inconsistency, ii, the

978-83-60810-14-9/08/$25.00 © 2008 IEEE

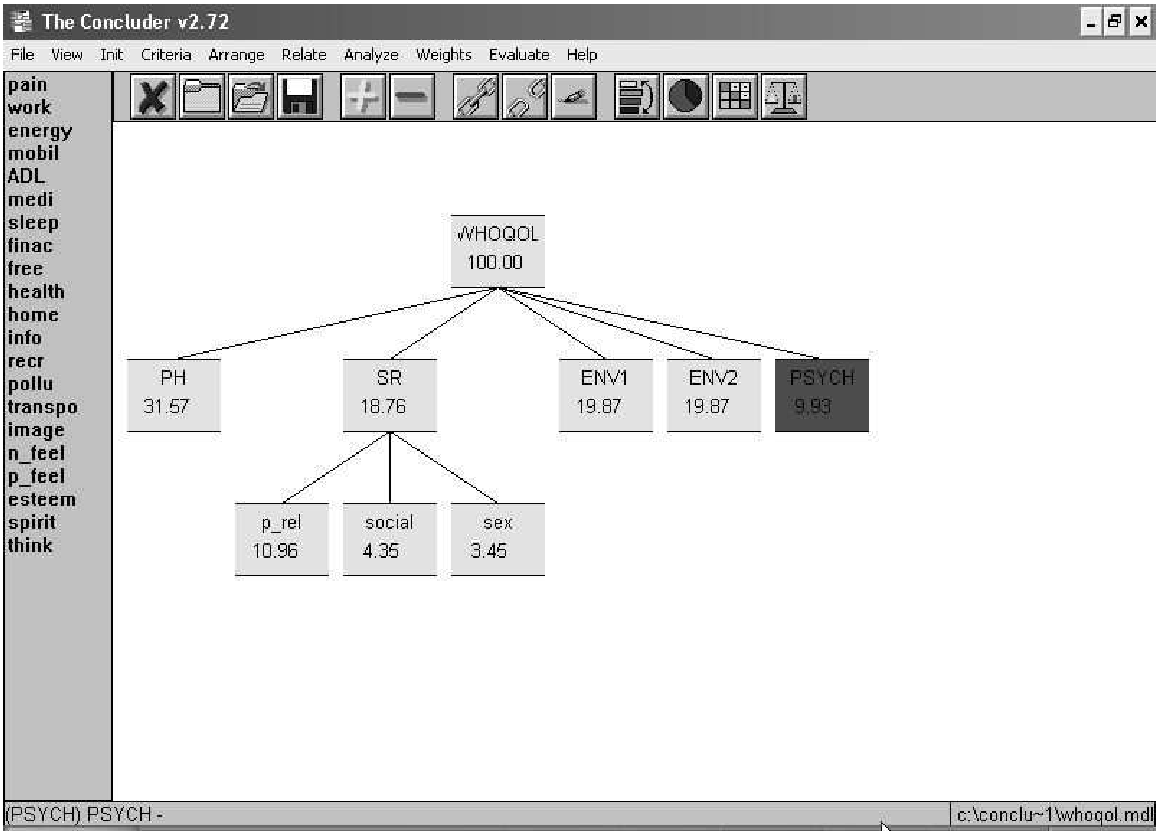


Fig. 1. Tree strusture

TABLE I
COMPARISONS FOR THE FIRST GROUP LEVEL

| | | |
|---|---|---|
| **1** | 2 | 4 |
| $\frac{1}{2}$ | **1** | 1 |
| $\frac{1}{4}$ | 1 | **1** |

TABLE II
WEIGHTS FOR SUBGROUP SR

| Group | Weight |
|---|---|
| A | 0.1096 |
| B | 0.0435 |
| C | 0.0335 |

following formula where: $ii$ =min($|1 - a_{ij}/(a_{ik} * a_{kj})|, |1 - a_{ik} * a_{kj}/a_{ij}|$) for $i = 1, j = 2$, and $k = 3$ (as introduced in [3]). was 0.5 and it was higher than the assumed threshold 1/3. The computed weights (as normalized geometric means of rows) are presented in Table 2. By changing, 4 to 3 for the illustration (rather than clinical), the new inconsistency index ii is computed as 1/3. Alternatively, another approach could have been changing $a_{33}$ to 2 resulting in $ii = 0$. purposes We stress that the changes here have been done for the illustration of the method but in real life, there must be a reason coming from the refined clinical knowledge.

As explained Koczkodaj [[3]], the above values were computed as normalized geometric means of the matrix rows. Figure 1 shows one highlighted subgroup, SR (social relationships), because attaching more subgroups creates a structure that requires additional space. Alternatively, we could show one group in one figure but again, five screen images, which are nearly identical would only make this presentation excessively long so the reader is asked to use his/her imagination as we enter "behind the scene" all objects (listed as unchained on the left hand side margin in Fig. 1) and compare them against each other assigning relative importance and paying attention to inconsistency as it was demonstrated for SR subgroup. Again, the comparisons have been done to illustrate the method, not the real instrument and the overall results for all criteria is presented in Table 3.

## CONCLUSION

Although the method of pairwise comparisons was originally used over 200 years ago, it has not been used to refine the properties of quality of life instruments. The method has strengthened the WHOQOL instrument by adding weights to individual items. Evidently, not all objects on the WHOQOL instrument are of equal importance. Appreciation of their relative differences, adds to the measure's precision. The inconsistency analysis further strengthens the measure by bringing the most problematic but often crucial comparisons of the instrument items. A challenge to the multiple experts in this tool's development can be "averaging" their individual assessments in the assumed model. Clinical trials and statistical analysis

TABLE III
THE FINAL WEIGHTS

| Object | Weight |
|---|---|
| p_rel | 0.1096 |
| pain | 0.1037 |
| finac | 0.0497 |
| free | 0.0497 |
| health | 0.0497 |
| home | 0.0497 |
| info | 0.0497 |
| recr | 0.0497 |
| pollu | 0.0497 |
| transpo | 0.0497 |
| social | 0.0435 |
| mobil | 0.0395 |
| work | 0.0369 |
| ADL | 0.0352 |
| medi | 0.0352 |
| sex | 0.0345 |
| energy | 0.0335 |
| sleep | 0.0318 |
| image | 0.0166 |
| n_feel | 0.0166 |
| p_feel | 0.0166 |
| esteem | 0.0166 |
| spirit | 0.0166 |
| think | 0.0166 |

need to follow the model enhancement. The enhancement to the WHOQOL project may be a challenging undertaking for years to come. Refinement of the WHOQOL may improve organizations' (such as WHO, the United Nations Educational, Scientific and Cultural Organization (UNESCO), United Nations) understanding of as well as health care professional practices in their efforts to assess quality of life.

## ACKNOWLEDGMENT

This work has been supported by the Ministry of Science and Higher Education, Republic of Poland, under project number N516 007 32/0978 decision No. 0978/T02/2007/32 in Poland, the NSERC grant and LURF in Canada.

## REFERENCES

[1] WHOQOL User's Manual, www.who.int/mental_health/media/en/76.pdf

[2] Kakiashvili, T., Kielan, K., Koczkodaj, W.W., Passi, K., Tadeusiewicz, R., Supporting the Asperger Syndrome Diagnostic Process by Selected AI Methods, proceedings of artificial intelligence studies. Vol. 4, pp. 21–27, 2007.

[3] Koczkodaj, W. W., A new definition of consistency of pairwise comparisons, Mathematical and computer modelling, (18)7, pp. 79–84, 1993.

[4] Koczkodaj, W. W., Herman, M. W., Orlowski, M. Using Consistency-driven Pairwise Comparisons in Knowledge-based Systems, International Conference on Information and Knowledge Management, 1997.

[5] Condorcet, M., The Essay on the Application of Analysis to the Probability of Majority Decisions, Paris: Imprimerie Royale, 1785.

[6] Fechner, G., Elemente der Psychophysik (1860, 2nd ed., 1889)

[7] Thurstone, L. L. (1927). A law of comparative judgement. Psychological Review, 34, 278–286.

[8] Saaty, T. L. (1977). A scaling method for priorities in hierarchical structures. Journal of Mathematical Psychology, 15, 234–281.

[9] McDowell. I., Measuring health : a guide to rating scales and questionnaires, 3rd ed., New York : Oxford University Press, 2006, 748 p.

## APPENDIX A BASIC CONCEPTS OF PAIRWISE COMPARISONS

An $n$ by $n$ pairwise comparisons matrix is defined as a square matrix $A = [a_{ij}]$ such that $a_{ij} > 0$ for every $i, j = 1, ..., n$. Each $a_{ij}$ expresses a relative preference of criterion (or stimulus)$s_i$ over criterion $s_j$ for $i, j = 1, ..., n$ represented by numerical weights (positive real numbers) and $w_i$ and $w_j$ respectively. The quotients $a_{ij} = w_i/w_j$ form a pairwise comparisons matrix:

$$A = \left| \begin{array}{cccc} 1 & a_{13} & ... & a_1 \\ & & & n \\ 1/a_1 & 1 & ... & a2 \\ 3 & & & n \\ ... & ... & ... & ... \\ 1/a_1 & 1/a_2 & ... & 1 \\ n & n & & \end{array} \right|$$

A pairwise comparisons matrix $A$ is called *reciprocal* if $a_{ij} = 1/a_{ji}$ for every $i, j = 1, ..., n$ (then automatically $a_{ii} = 1$ for every $i = 1, ..., n$ because they represent the relative ratio of a criterion against itself). A pairwise comparisons matrix $A$ is called *consistent* if $a_{ij} \cdot a_{jk} = a_{ik}$ holds for every $i, j, k = 1, ..., n$ since $w_i/w_j \cdot w_j/w_k$ is expected to be equal to $w_i/w_k$. Although every consistent matrix is reciprocal, the converse is not generally true. In practice, comparing of $s_i$ to $s_j$, $s_j$ to $s_k$, and $s_i$ to $s_k$ often results in inconsistency amongst the assessments in addition to their inaccuracy; however, the inconsistency may be computed and used to improve the accuracy.

The first step in pairwise comparisons is to establish the relative preference of each combination of two criteria. A scale from 1 to 5 can be used to compare all criteria in pairs. Values from the interval $[1/5, 1]$ reflect inverse relationships between criteria since $s_i/s_j = 1/(s_j/s_i)$. The consistency driven approach is based on the reasonable assumption that by finding the most inconsistent judgments, one can then reconsider one's own assessments. This in turn contributes to the improvement of judgmental accuracy. Consistency analysis is a dynamic process which is assisted by the software.

The central point of the inference theory of the pairwise comparisons is Saaty's Theorem, [8], which states that for every $n$ by $n$ consistent matrix $A = [a_{ij}]$ there exist positive real numbers $w_1, ..., w_n$ (weights corresponding to criteria $s_1, .n.., s)$ such that $a_{ij} = w_i/w_j$ for every $i, j = 1, ..., n$. The weights $w_i$ are unique up to a multiplicative constant. Saaty (1977) also discovered that the eigenvector corresponding to the largest eigenvalue of $A$ provides weights $w_i$ which we wish to obtain from the set of preferences $a_{ij}$. This is not the only possible solution to the weight problem. In the past, a least squares solution was known, but it was far more computationally demanding than finding an eigenvector of a matrix with positive elements. Later, a method of row geometric means was proposed (Jensen, 1984), which is the simplest and most effective method of finding weights. A statistical experiment demonstrated that the accuracy, that is, the distance from the original matrix $A$ and the matrix $AN$

reconstructed from weights with elements $[a_{ij}] = [w_i/w_j]$, does not strongly depend on the method. There is, however, a strong relationship between the accuracy and consistency. Consistency analysis is the main focus of the consistency driven approach.

An important problem is how to begin the analysis. Assigning weights to all criteria (e.g., $A = 18, B = 27, C = 20, D = 35$) seems more natural than the above process. In fact it is a recommended practice to start with some initial values. The above values yield the ratios: $A/B = 0.67$, $A/C = 0.9$, $A/D = 0.51$, $B/C = 1.35$, $B/D = 0.77$, $C/D = 0.57$. Upon analysis, these may look somewhat suspicious because all of them round to 1, which is of equal or unknown importance. This effect frequently arises in practice, and experts are tempted to change the ratios by increasing some of them and decreasing others (depending on knowledge of the case). The changes usually cause an increase of inconsistency which, in turn, can be handled by the analysis because it contributes to establishing more accurate and realistic weights. The pairwise comparisons method requires evaluation of all combinations of pairs of criteria, and can be more time consuming because the number of comparisons depends on $n^2$ (the square of the number of criteria). The complexity problem has been addressed and partly solved by the introduction of hierarchical structures [8]. Dividing criteria into smaller groups is a practical solution in cases in which the number of criteria is large.

## Appendix B Consistency Analysis

Consistency analysis is critical to the approach presented here because the solution accuracy of *not-so-inconsistent* matrices strongly depends on the inconsistency. The consistency driven approach is, in brief, the next step in the development of pairwise comparisons.

The challenge to the pairwise comparisons method comes from a lack of consistency in the pairwise comparisons matrices which arises in practice. Given an $n$ by $n$ matrix $A$ that is not consistent, the theory attempts to provide a consistent $n$ by $n$ matrix $AN$ that differs from matrix $A$ "as little as possible". In particular, the geometric means method produces results similar to the eigenvector method (to high accuracy) for the ten million cases tested. There is, however, a strong relationship between accuracy and consistency.

Unlike the old eigenvalue based inconsistency, introduced in [8], the triad based inconsistency locates the most inconsistent triads [3]. This allows the user to reconsider the assessments included in the most inconsistent triad.

Readers might be curious, if not suspicious, about how one could arrive at values such as 1.30 or 1.50 as relative ratio judgments. In fact the values were initially different, but have been refined and the final weights have been calculated by the consistency analysis. It is fair to say that making comparative judgments of rather intangible criteria (e.g., overall alteration and/or mineralization) results not only in imprecise knowledge, but also in inconsistency in our own judgments. The improvement of knowledge by controlling inconsistencies in the judgments of experts, that is, the consistency driven approach, is not only desirable but is essential.

In practice, inconsistent judgments are unavoidable when at least three factors are independently compared against each other. For example, let us look closely at the ratios of the four criteria $A$, $B$, $C$, and $D$ in Figure $C1$. Suppose we estimate ratios $A/B$ as 2, $B/C$ as 3, and $A/C$ as 5. Evidently something does not add up as $(A/B)@\ (B/C) = 2 \cdot 3 = 6$ is not equal to 5 (that is $A/C$). With an inconsistency index of 0.17, the above triad (with highlighted values of 2, 5, and 3) is the most inconsistent in the entire matrix (reciprocal values below the main diagonal are not shown in Figure $C1$). A rash judgment may lead us to believe that $A/C$ should indeed be 6, but we do not have any reason to reject the estimation of $B/C$ as 2.5 or $A/B$ as 5/3. After correcting $B/C$ from 3 to 2.5, which is an arbitrary decision usually based on additional knowledge gathering, the next most inconsistent triad is (5,4,0.7) with an inconsistency index of 0.13. An adjustment of 0.7 to 0.8 makes this triad fully consistent ($5 \cdot 0.8$ is 4), but another triad (2.5,1.9,0.8) has an inconsistency of 0.05. By changing 1.9 to 2 the entire table becomes fully consistent. The corrections for real data are done on the basis of professional experience and case knowledge by examining all three criteria involved.

An acceptable threshold of inconsistency is 0.33 because it means that one judgment is not more than two grades of the scale 1 to 5 away (an off-by-two error) from the remaining two judgments. There was no need to continue decreasing the inconsistency, as only its high value is harmful; a very small value may indicate that the artificial data were entered hastily without reconsideration of former assessments.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 155–159

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Accuracy Boosting Induction of Fuzzy Rules with Artificial Immune Systems

Adam Kalina
Value Based Advisors Sp. z o.o.
ul. Połabian 35, 52-339 Wrocław, Poland
Adam.Kalina@vba.pl

Edward Mężyk
and Olgierd Unold
Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
Wyb. Wyspianskiego 27, 50-370 Wrocław, Poland
Olgierd.Unold@pwr.wroc.pl

***Abstract*—The paper introduces accuracy boosting extension to a novel induction of fuzzy rules from raw data using Artificial Immune System methods. Accuracy boosting relies on fuzzy partition learning. The modified algorithm was experimentally proved to be more accurate for all learning sets containing non-crisp attributes.**

## I. Introduction

FUZZY-BASED data mining is a modern and very promising approach to mine data in an efficient and comprehensible way. Moreover, fuzzy logic [8] can improve a classification task by using fuzzy sets to define overlapping class definitions. This kind of data mining algorithms discovers a set of rules of the form “IF (fuzzy conditions) THEN (class),” whose interpretation is as follows: IF an example’s attribute values satisfy the fuzzy conditions THEN the example belongs to the class predicted by the rule. The automated construction of fuzzy classification rules from data has been approached by different techniques like, e.g., neuro-fuzzy methods, genetic-algorithm based rule selection, and fuzzy clustering in combination with other methods such as fuzzy relations and genetic algorithm optimization (for references see [10]).

A quite novel approaches, among others, integrate Artificial Immune Systems (AISs) [3] and Fuzzy Systems to find not only accurate, but also linguistic interpretable fuzzy rules that predict the class of an example. The first AIS-based method for fuzzy rules mining was proposed in [2]. This approach, called IFRAIS (Induction of Fuzzy Rules with an Artificial Immune System), uses sequential covering and clonal selection to learn IF-THEN fuzzy rules. In [7] the speed of IFRAIS was improved significantly by buffering dicovered fuzzy rules in a clonal selection. One of the AIS-based algorithms for mining IF-THEN rules is based on extending the negative selection algorithm with a genetic algorithm [4]. Another one is mainly focused on the clonal selection and so-called a boosting mechanism to adapt the distribution of training instances in iterations [1]. A fuzzy AIS was proposed also in [6], however that work addresses not the task of classification, but the task of clustering.

This paper seeks to boost an accuracy of IFRAIS approach by exploring the use of fuzzy partitions learning.

## II. IFRAIS

Data preparation for learning in IFRAIS consists of the following steps: (1) create a fuzzy variable for each attribute in data set; (2) create class list for actual data set; (3) and compute information gain for each attribute in data set.

Listing 1. Sequential covering algorithm

```
Input: full training set
Output: fuzzy rules set

rules set = 0
FOR EACH class value c in class values list DO
  values count = number of c in full training set
  training set = full training set
  WHILE values count > number of maximal uncovered
      examples AND
    values count > percent of maximal uncovered
      examples
    rule = CLONAL–SELECTION–ALGORITHM(training set,
      c)
    covered  = COVER–SET(training set, rule)
    training set = training set / covered with rule
      set
    values count = values count – size of covered
    ADD(rules set, rule)
  END WHILE
END FOR EACH
training set = full training set
FOR EACH rule R in rules set DO
  MAXIMIZE–FITNESS(R, training set)
  COMPUTE–FITNESS(R, training set)
END FOR EACH
RETURN rules set
```

IFRAIS uses a sequential covering as a main learning algorithm (see Listing 1). In the first step a set of rules is initialized as an empty set. Next, for each class to be predicted the algorithm initializes the training set with all training examples and iteratively calls clonal selection procedure with the parameters: the current training set and the class to be predicted. The clonal selection procedure returns a discovered rule and next the learning algorithm adds the rule to the rule set and removes from the current training set the examples that have been correctly covered by the evolved rule.

Clonal selection algorithm is used to induct rule with the best fitness from training set (see Listing 2). Basic elements of this method are antigens and antibodies which refers directly to biological immune systems. Antigen is an example from

978-83-60810-14-9/08/$25.00 © 2008 IEEE

data set and antibody is a fuzzy rule. Similarly to fuzzy rule structure, which consists of fuzzy conditions and class value, antibody comprises genes and informational gene. Number of genes in antibody is equal to number of attributes in data set. Each gene consists of a fuzzy rule and an activation flag that indicates whether fuzzy condition is active or inactive.

Listing 2. Clonal selection algorithm in IFRAIS (based on [2])

```
Input: training set, class value (c)
Output: fuzzy rule

CREATE randomly antibodies population with size s
    and class value c
FOR EACH antibody A in antibodies population
  PRUNE(A)
  COMPUTE-FITNESS(A, training set)
END FOR EACH
FOR i=1 TO number of generations n DO
  WHILE clones population size < s-1
    antibody to clone = TOURNAMENT-SELECTION(
        antibodies population)
    clones = CREATE x CLONES(antibody to clone)
    clones population = clones population + clones
  END WHILE
  FOR EACH clone K in clones population
    muteRatio = MUTATION-PROBABILITY(K)
    MUTATE(K, muteRatio)
    PRUNE(K)
    COMPUTE-FITNESS(K, training set)
  END FOR EACH
 antibodies population = SUCCESSION(antibodies
    population, clones population)
END FOR
 result = BEST-ANTIBODY(antibodies population)
RETURN result
```

In the first step the algorithm generates randomly antibodies population with informational gene equals to class value *c* passed in algorithm parameter. Next each antibody from generated population is pruned. Rule pruning has a twofold motivation: reducing the overfitting of the rules to the data and improving the simplicity (comprehensibility) of the rules [11]. Fitness of the rule is computed according to the formula

$$FITNESS(rule) = \frac{TP}{TP+FN} \cdot \frac{TN}{TN+FP} \tag{1}$$

where TP is number of examples satisfying the rule and having the same class as predicted by the rule; FN is the number of examples that do not satisfy the rule but have the class predicted by the rule; TN is the number of examples that do not satisfy the rule and do not have the class predicted by the rule; and FP is the number of examples that satisfy the rule but do not have the class predicted by the rule. Since the rules are fuzzy, the computation of the TP, FN, TN and FP involves measuring the degree of affinity between the example and the rule. This is computed by applying the standard aggregation fuzzy operator *min*

$$AFFINITY(rule, example) = min_{i=1}^{condCount}(\mu_i(att_i)) \tag{2}$$

where $\mu_i(att_i)$ denotes the degree to which the corresponding attribute value $att_i$ of the example belongs to the fuzzy set accociated with the *i*th rule condition, and *condCount* is the number of the rule antecedent conditions. The degree of membership is not calculated for an inactive rule condition, and if the *i*th condition contains a negation operator, the membership function equals to $(1 - \mu_i(att_i))$ (complement). An example satisfies a rule if $AFFINITY(rule, example) > L$, where $L$ is an activation threshold. Next, antibody to be cloned is selected by tournament selection from the antibodies population. For each antibody to be cloned the algorithm produces *x* clones. The value of *x* is proportional to the fitness of the antibody. Next, each of the clones undergoes a process of hypermutation, where the mutation rate is inversely proportional to the clone's fitness. Once a clone has undergone hypermutation, its corresponding rule antecedent is pruned by using the previously explained rule pruning procedure. Finally, the fitness of the clone is recomputed, using the current training set. In the last step the *T*-worst fitness antibodies in the current population are replaced by the *T* best-fitness clones out of all clones produced by the clonal selection procedure. Finally, the clonal selection procedure returns the best evolved rule, which will then be added to the set of discovered rules by the sequential covering. More details of the IFRAIS is to be found in [2].

## III. Fuzzy partition inference

IFRAIS, as an Artificial Immune System evolves a population of antibodies representing the IF part of a fuzzy rule, whereas each antigen represents an example. As was stated, each rule antecedent consists of a conjunction of rule condition. In IFRAIS approach three and only three linguistic terms (low, medium, high) are associated with each continuous attribute. Each linguistic term is represented by the triangular membership functions (see Fig 1). It seems to be purposeful to infer a fuzzy partition for each continuous attribute over the data set instead of stiff, and the same for different attributes partitioning. There exist various methods to learn a fuzzy partions over a set of data [5]. We consider using a clonal selection algorithm to automatic infer partitions for each attribute, both crisp and continuous one. In such an approach a population of antibodies represent a set of partitions, and an antigen is a whole set of data.

### A. Representation

Each partition is represented by two crisp tables: a size table $S$ and a range table $R$. $S$ is one-dimensional table $S = \{s_1, s_2, \ldots, s_{setsCount}\}$, where $setsCount$ is the number of fuzzy sets the partition is consisted of (the number of ligustic terms). $setCount$ is drawn from the range $[3, 12]$ during inference. $s_i$ is a size of a fuzzy set and is expressed in so-called "‘division points"’ $pv$. For a crisp attribute $pv$ corresponds to one attribute value. For an attribute to be fuzzified $pv$ is computed as follows

$$pv = \frac{maxValue - minValue}{apc} \tag{3}$$

where $maxValue$ is a maximal attribute value, $minValue$ is a minimal attribute value, and $apc = setsCount \cdot pc$ defines the number of $pv$-points allocated to the attribute. $pc$ is an algorithm parameter set to 10, and $pc$ is interpreted as a number of $pv$-points assigned to one fuzzy set of the partition. While fuzzy partition infering, the size of one fuzzy set $s_i$ is drawn from a range $[1, apc - apc \cdot apcp]$, where $apcp$ is a percent of $apc$ ($apcp$ is a program parameter set to 10%).

A range table $R$ is two-dimensional table contains pairs $R = \{(r_{l_1}, r_{r_1}), (r_{l_2}, r_{r_2}), \ldots, (r_{l_{setsCount}}, r_{r_{setsCount}})\}$, where a pair $(r_{l_i}, r_{r_i})$ is a range expressed in $pv$ for which triangle fuzzy set (linguistic term) will be created. $r_{l_i}$ is a lower bound and $r_{r_i}$ is upper bound of the $i$th range. For crisp attributes each cell of the $R$ table contains induced list of attribute values. For example, partition of unfuzzy attribute FALLOUT, could be divided to 3 fuzzy sets $S = \{1, 1, 2\}$ with following value list $R = \{(snow), (rain), (hail, nofallout)\}$. The range for triangle fuzzy set is calculated on the basis of a size of fuzzy set $s_i$, $pv$, $minValue$, $maxValue$, $r_{r_{i-1}}$, and point overflow index $oi$. $oi$ is calculated as a difference between a sum of all values in $S$ table and $apc$. In the case when $oi > 0$, an overalap index $ovi$ is randomly generated from the range $[1, oi]$ to prevent lack of attribute full covering by lingustic terms. If $oi > r_{r_{i-1}}$, then $ovi$ is drawn from the range $[1, r_{r_{i-1}}]$.

To the first lower bound $r_{l_0}$ is assigned $minValue$. The first upper bound $r_{r_0}$ equals the size of a fuzzy set $s_1$ multiplied by $pv$ value. Every next pair in the $R$ table is calculated according to the following rules: if $oi > 0$ then draw $ovi$ from $[1, oi]$ or from $[1, r_{r_{i-1}}]$ if $oi > r_{r_{i-1}}$. Otherwise $ovi$ is equal to 0. The lower bound $r_{l_i} = ovi \cdot pv$, whereas the upper bound $r_{r_i} = r_{l_i} + s_i \cdot pv$. If $r_{r_i} > maxValue$ then $r_{r_i}$ is truncated to $maxValue$. Figure 2 illustrates the one of the fuzzy partion learning steps for an attribute TIME. The values from $S = \{15, 20, 5, 10, 25\}$ are represented by double side arrows. $ovi$ values are represented by one side arrows. $R$ table goes as follows $R = \{(0, 15), (5, 25), (23, 28), (25, 35), (25, 50)\}$.

### B. Operations

Fuzzy variables ready to use by IFRAIS are created from the range table $R$. All partition modifications are made over the size table $S$, from which table $R$ is derived according to the mentioned above rules. Fuzzy partitions are evaluated by using following operations:

*strong splitting*—removes randomly number $\chi$ of $pv$-points from randomly chosen cell $s_i$, and creates at randomly chosen position in $S$ new cell including $\chi$-points,

*light splitting*—removes only one $pv$-point form randomly chosen cell $s_i$, and creates at randomly chosen position in $S$ new cell including one point,

*strong joining*—removes randomly number $\chi$ of $pv$-points from randomly chosen cell $s_i$, and adds $\chi$-points to the randomly chosen cell,

*light joining*—removes only one $pv$-point from randomly chosen cell $s_i$, and adds one points to the randomly chosen cell.

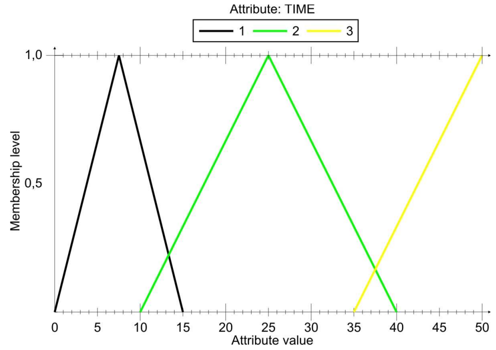


Fig. 1. An example of fuzzy partition for TIME attribute

### C. Inferring

Fuzzy partition inferring is based on clonal selection algorithm (see Listing 3). In this algorithm it is worth underlining a clone mutation which undergoes for all clones in a population. While mutating, only the sizes of fuzzy sets (table $S$) for choosen attribute are generated according to the mentioned above rules, but without changing the number of liguistic terms (the size of table $S$). After a new partition generation, the operations (strong and light splitting, strong and light joining) are fired, each with the probability $muteRatio$

$$muteRatio = min + \frac{(max - min) \cdot (1 - f + No)}{2} \quad (4)$$

where $min$ and $max$ are minimal and maximal probability respectively (algorithm parameters), $No$ is a number taken at random from $[0, 1]$, and $f$ is a normalized clone fitness before mutation.

Listing 3. Fuzzy partition learning

```
Input: training data set, IFRAIS system parameters,
    algorithm parameters
Output: fuzzy partitions set

Randomly generate antibodies population of size s
FOR EACH antibody A in antibodies population
  COMPUTE–FITNESS(A, training data set)
END FOR EACH
FOR i=1 TO number of generations n DO
  SORT–DESCENDING(antibodies population)
  WHILE clones population size < maximal clones
      population size
    antibody to clone = TAKE–NEXT–BEST(antibodies
        population)
    clones = CREATE x CLONES(antibody to clone)
    clones population = clones population + clones
  END WHILE
  FOR EACH clone K in clones population
    MUTATE (K)
    COMPUTE–FITNESS(K, training data set)
  END FOR EACH
  Antibodies population = SUCCESION(antibodies
      population, clones population)
END FOR
result = BEST–ANTIBODY(antibodies population)
RETURN result
```

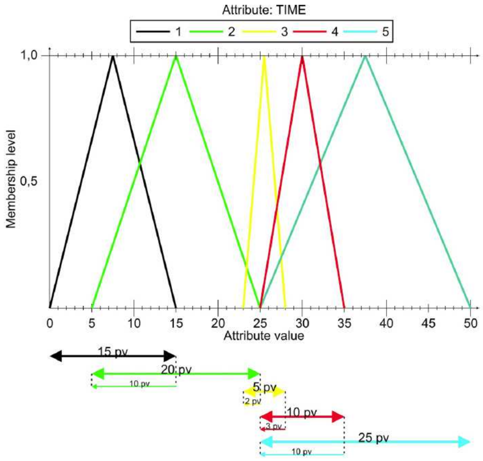


Fig. 2. Induced fuzzy partition for TIME attribute

TABLE I
DATA SETS AND NUMBER OF ROWS, ATTRIBUTES, CONTINUOUS ATTRIBUTES, AND CLASSES

| Data set | #Rows | #Attrib. | #Cont. | #Class. |
|---|---|---|---|---|
| Bupa | 345 | 6 | 6 | 2 |
| Crx | 653 | 15 | 6 | 2 |
| Hepatitis | 80 | 19 | 6 | 2 |
| Ljubljana | 277 | 9 | 9 | 2 |
| Wisconsin | 683 | 9 | 9 | 2 |
| Votes | 232 | 16 | 0 | 2 |

## IV. EXPERIMENTAL RESULTS

In order to evaluate the performance of the speed boosting extensions, both IFRAIS and improved IFRAIS were applied to 6 public domain data sets available from the UCI repository (http://archive.ics.uci.edu/ml/datasets.html):

- Bupa (Liver+Disorders)
- Crx (Credit+Approval)
- Hepatitis (Hepatitis)
- Lubljana (Breast+Cancer)
- Votes (Congressional+Voting+Records)
- Winsconsin (Breast+Cancer+Wisconsin+(Original))

The experiments were conducted using a Distribution-Balanced Stratified Cross-Validation [12], which is a one of the version of well-known $k$-fold cross-validation, and improves the estimation quality by providing balanced intraclass distributions when partitioning a data set into multiple folds. Additionally, both IFRAIS method were compared to C4.5, well-known data mining algorithm for discovering classification rules [9] (results of C4.5 taken from [2]).

Table 1 shows the number of rows, attributes, continuous attributes, and classes for each data set. Note that only continuous attributes are fuzzified. The Votes data set does not have any continuous attribute to be fuzzified, whereas the other data sets have 6 or 9 continuous attributes that are fuzzified by IFRAIS. All experiments with IFRAIS were repeated 50-times using 5-fold cross-validation. Table 2 shows for each data set the average accuracy rate with standard deviations, both for IFRAIS and IFRAIS extended by fuzzy partition learning (boosted IFRAIS), as well as for C4.5 method. As shown in Table 2 the boosted IFRAIS obtained better accuracy rates than standard IFRAIS for all data set but one (the Votes data set comprises only crisp attributes). For Bupa set the average gain is ca 14 %, and for Hepatitis even more (16.6 %)! The improved IFRAIS obtained higher accuracy than C4.5 in five out of the six data sets. C4.5 obtained a higher accuracy than IFRAIS in only one data set (Crx), but the differences in accuracy rate is not significant, since the accuracy rate intervals (based on the standard deviations) overlap.

TABLE II
ACCURACY RATE ON THE TEST SET

| Data set | C4.5 | IFRAIS | Boosted IFRAIS |
|---|---|---|---|
| Bupa | 67.40±1.60 | 58.38±0.78 | 72.34±0.99 |
| Crx | 90.22±1.59 | 86.03±0.26 | 87.37±0.08 |
| Hepatitis | 76.32±2.79 | 77.25±1.71 | 93.87±2.65 |
| Ljubljana | 68.80±4.45 | 69.55±1.09 | 74.70±0.79 |
| Wisconsin | 95.32±1.09 | 94.91±0.39 | 97.39±0.28 |
| Votes | 94.82±0.82 | 96.98±0.00 | 96.98±0.00 |

## V. CONCLUSION

The accuracy boosting extension was introduced to the IFRAIS algorithm—an AIS-based method for fuzzy rules mining. Boosting uses the fuzzy partion learning based on the clonal selection. The partition inferring improves significantly effectiveness of an algorithm. Although the proposed improvements increase the accuracy of the whole algorithm, it is worth noticing that the learning of fuzzy partition is additional time-consuming procedure.

It seems to be still possible to improve the Induction of Fuzzy Rules with Artificial Immune Systems, and not only considering the effectiveness of the induced fuzzy rules but also time of working. These two goals could be achieved mostly by modifying the fitness function to reinforce the fitness of high-accuracy rules, as in [1]. We also consider changing the triangular membership functions to various more sophisticated functions and manipulating all system parameters to obtain higher quality results. We have performed preliminary experiments, in which speed and accuracy boosting IFRAIS with modified fitness function and parameters is trained on well known data sets. The results are very promising.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Robert Alves for making available to them his source code of IFRAIS.

## REFERENCES

[1] Alatas, B., Akin, E.: Mining Fuzzy Classification Rules Using an Artificial Immune System with Boosting. In: *Eder, J. et al. (eds.) ADBIS 2005.* LNCS, vol. 3631, pp. 283–293. Springer-Verlag Berlin Heidelberg (2005).

[2] Alves, R. T., et al.: An artificial immune system for fuzzy-rule induction in data mining. In: *Yao, X., et al (eds.) Parallel Problem Solving from Nature—PPSN VIII.* LNCS, vol. 3242, pp. 1011–1020. Springer Heidelberg (2004).

[3] Dasgupta, D.(ed.): Artificial Immune Systems and Their Applications. Spring-Verlag Berlin Heidelberg Germany (1999).

[4] Gonzales, F. A., Dasgupta, D.: An Immunogenetic Technique to Detect Anomalies in Network Traffic. In: *Proceedings of Genetic and Evolutionary Computation.* pp. 1081–1088. Morgan Kaufmann San Mateo (2002).

[5] Marsala C.: Fuzzy Partitioning Methods, Granular Computing: An Emerging Paradigm. Physica-Verlag GmbH Heidelberg Germany, pp. 163–186 (2001).

[6] Nasaroui, O., Gonzales, F., Dasgupta, D.: The Fuzzy Artificial Immune System: Motivations, Basic Concepts, and Application to Clustering and Web Profiling. In: *Proceedings of IEEE International Conference on Fuzzy Systems,* pp. 711–716 (2002).

[7] Mężyk E., Unold O.: Speed Boosting Induction of Fuzzy Rules with Artificial Immune Systems. In: *Mastorakis, E. M. et al. (eds) Proc. of the 12th WSEAS International Conference on SYSTEMS,* Heraklion Greece July 22-24, pp. 704–706 (2008).

[8] Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. Analysis and Design. MIT Press Cambridge (1998).

[9] Quinlan, J. R.: C4.5: Programs For Machine Learning. Morgan Kaufmann San Mateo (1993).

[10] Roubos J. A., Setnes M. Abonyi J.: Learning fuzzy classification rules from labeled data. Information Science 150, pp. 77–93 (2003).

[11] Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann San Mateo (2005).

[12] Zeng X., Martinez T. R.: Distribution-Balanced Stratified Cross-Validation for Accuracy Estimations. *Journal of Experimental and Theoretical Artificial Intelligence.* Vol. 12, number 1, pp. 1–12. Taylor and Francis Ltd (2000).

ISSN 1896-7094
©2008 PIPS

# On Classification Tools for Genetic Algorithms

Stefan Kotowski*†, Witold Kosiński*‡, Zbigniew Michalewicz*§, Piotr Synak* and and Łukasz Brocki*
*Faculty of Computer Science, Polish-Japanese Institute of Information Technology ul. Koszykowa 86 02-008 Warszawa, Poland
† Institute of Fundamental Technological Research
IPPT PAN, ul. Świętokrzyska 21, 00-049 Warszawa Poland
‡Institute of Environmental Mechanics and Applied Computer Science
Kazimierz Wielki University
ul. Chodkiewicza 30, 85-064 Bydgoszcz, Poland
§ University of Adelaide, School of Computer Sciences,
South Australia 5005, Australia

{skot, wkos, synak, lucas}@pjwstk.edu.pl, zbyszek@cs.adelaide.edu.au

***Abstract*—Some tools to measure convergence properties of genetic algorithms are introduced. A classification procedure is proposed for genetic algorithms based on a conjecture: the entropy and the fractal dimension of trajectories produced by them are quantities that characterize the classes of the algorithms. The role of these quantities as invariants of the algorithm classes is discussed together with the compression ratio of points of the genetic algorithm.**

## I. Introduction

THERE is the so-called "No-free lunch theorem" [10] algorithms and moreover, one cannot find most suitable operator according to which: it does not exist a best evolutionary between all possible mechanisms of crossover, mutation and selection without referring to the particular class of optimisation problems under investigation. Evolutionary algorithms are the methods of optimizations which use a limited knowledge about investigated problem. On the other hand, our knowledge about the algorithm in use is often limited as well [11], [12].

One of the most difficult, however, of practical importance, problems is the choice of an algorithm to given optimisation problem.

The distinguishing between optimisation problem and the algorithm (its choice) leads to the main difficulty. Consequently, the distinguishing is an artificial operation because it abstains from the idea of genetic algorithm (GA), since the fitness function, arises from the cost function (i.e. the function to be optimised) is the main object of the genetic algorithm and it emerges form the formulation of the optimisation problem and it is difficult to speak on genetic algorithm as an operator without the fitness function. However, in our consideration we will simultaneously use the both notions of the genetic algorithms. The first notion as an operator acting on the cost (fitness) function, the second—as a specific (real) algorithm for which the fitness is the main component being the algorithm's parameter.

This dual meaning of the genetic algorithm is crucial for our consideration, because our main aim is to try to classify genetic algorithms. The classification should lead to a specific choice methodology of genetic algorithms understood as operators. It is expected that in terms of this methodology one will be able to choose the appropriate algorithm to given optimisation problem. We aim that using this classification one could improve existing heuristic methods of assortment of genetic algorithms which are based mainly on experiences and programmer intuition.

During the action of genetic algorithm several random operations are performed and each action generates a sequence of random variables, which are candidate solutions. The action of GA could be represented in the solution (or rather—search) space as a random trajectory.

The present paper is an attempt to introduce an enlarged investigation method to the theory of genetic (evolutionary) algorithms. We aim at the development of some tools suitable for characterization of evolutionary algorithms based on the notions of the symbolic dynamics as well as on compression rates.

## II. Classification of algorithms and its tools

The convergence of GAs is one of the main issues of the theoretical foundations of GAs, and has been investigated by means of Markov's chains. The model of GA as a Markov's chain is relatively close to the methods known in the theory of dynamical systems.

In the analysis of GAs regarded as (stochastic) dynamical systems one can use the fact, (proven by Ornstein and Friedman [3], [8]) which state that mixing Markov's chains are Bernoulli's systems and consequently, the entropy of the systems is a complete metric invariant.

Those facts enable us to classify GAs using the entropy. The systems for which the entropies have the same value are isomorphic. Hence the entropy makes it possible to classify GAs by splitting them into equivalence classes. Unfortunately, in many cases the determination of the entropy is very difficult or even impossible. Hence one can try to use as classification

978-83-60810-14-9/08/$25.00 © 2008 IEEE

tools the compression rate and the fractal dimension determined for trajectories of GAs.

## III. Genetic algorithms

Let $X$ be a space of solutions of an optimisation problem characterized by a fitness function $f : X \to \mathbf{R}, X \subset \mathbf{R}^l$ for which a genetic algorithm will be invented. Each element $x \in X$ will be encoded in the form of a binary chromosome of the length $N$. The coding function $\varphi : X \to \{0,1\}^N = B$ maps elements of $X$ into chromosome from the $B$ space.

Let us assume that the genetic algorithm acts on $K$-element populations. Each population forms a multiset $[P^K]$ in the product space $B^K$, for the $i$-th generation we will use the denotation $[P_i^K]$, for the population and each element of this multiset can be identified with a vector

$$P_i^K = [x_1^i, x_2^i, \dots, x_K^i] \tag{1}$$

rembering that a population is an equivalent class of points from the vector space $B^K$. The equivalent relation is defined by the class of all possible permutations of the set of $K$-th numbers $\{1, 2, ..., K\}$.

Let us notice that we can identify points from $X$ with their encoded targets in $B$ under the action of space $X^K$. By a trajectory of the genetic algorithm of the duration $M$ we mean a set

$$T = \bigcup_{i=1}^{M} [P_i^K], \tag{2}$$

where $M$ is the number of steps (generations) of the genetic algorithm which is realized.

Let $p_m$ and $p_c$ be the probabilities of the mutation and crossover, respectively, while $p_s$ is the probability of selection, all independent from the generation.

Then, for such a genetic algorithm the probability of the appearance of the population $[P_{i+1}^K]$ at the generation $i+1$ after the population $[P_i^K]$ at the generation $i$, is the conditional probability

$$\mathcal{P}(P_{i+1}^K | P_i^K \ , f(P_i^K), p_m, p_c, p_s). \tag{3}$$

Here by $f(P_i^K)$ we understand the vector–valued function $[f(x_1^i), f(x_2^i), \dots, f(x_K^i)]$. The initial population $[P_1^K]$ is generated by the use of a uniform probability distribution over the set $B$, i.e. each point from $B$ has the same probability of being selected as a member (component) of $[P_1^K]$. Next populations following that one, are the results of the action of the GA and, hence, may have a non-uniform probability distribution.

Let us notice that in view of our assumptions it follows from (3) that the probability of the appearance of each population depends on the previous population and does not depend on the history (i.e. on earlier population; the probabilities $p_m, p_c$ and $p_s$ can be regarded as parameters of the function $\mathcal{P}$).

Now, if we look at the trajectory of the GA defined by (2), we can see that its generation is an ergodic (mixing) process and Markov's one. Subsequent populations (i.e. points of the trajectory) are states of the process, about which we can say that each state is accessible with the probability 1.

## IV. Entropy of GA

The domain of research of the ergodic theory is a space with measure and mappings which preserve it. Let us denote by $\mathsf{T}_i$ the operator which maps $i$-th generation (point of the trajectory) into the next one. Having the probability distribution (3) characterizing the mapping $\mathsf{T}_i$ from one population to another, we can define the entropy of the mapping

$$H(\mathsf{T}_i) = -\sum_{j=1}^{M} \mathcal{P}(P_{i+1,j}^K | P_i^K, f(P_i^K), p_m, p_c, p_s) \\ \log\mathcal{P}(P_{i+1,j}^K | P_i^K, f(P_i^K), p_m, p_c, p_s) \tag{4}$$

where $[P_{i+1,j}^K]$ is a possible population from $B, j = 1, 2, \dots, 2^{NK}, \dots, M$.

According to our previous proposition the initial population is generated by the use of a uniform probability, and attains the maximal value generated by the GA. In the next step the probabilities of populations are not uniform and differ at each generation; this is the essence of the action of GA. Consequently the entropy of the mapping $\mathsf{T}_i$ decreases. In the limit case when the number of steps tends to infinity one could expect that the terminal population will be composed of $K$ copies (exactly speaking, according to (1) – a cartesian product) of the same element (an optimal solution). However, this case will be possible only in the case of the pointwise asymptotic stability of GA. In general, the entropy will tend to minimum.

Entropy is the function of the probability of mutation and selection; it grows with the growing mutation probability and decreases when the selection pressure grows. Then the entropy could realize a measure of interactions between mutations and selection operators. Entropy also depends on the number of elements in population and it is decreasing when the population grows. The entropy value of the trajectory could be linked with computational complexity of the evolutionary algorithms.

## V. Fractal dimensions

Since the determination of the probability of the mapping $\mathsf{T}_i$, as well as the entropy $H_i$, in an analytical way is rather difficult to be performed, we are proposing to substitute them with a fractal dimension which is related to the entropy [8] and can characterize non-deterministic features of GA. In [7] general statistical and topological methods of analysis of GAs have been introduced. Moreover one can use Hausdorff's dimension or its approximation as an invariant of equivalence of algorithms.

To be more evident, let us recall the notion of the $s$-dimensional Hausdorff measure ([4]) of the subset $E \subset \mathbf{R}^l$, where $s \geq 0$. If $E \subset \bigcup_i U_i$ and diameter of $U_i$, denoted by $\delta(U_i)$, is less than $\epsilon$ for each $i$, we say that $\{U_i\}$ is an $\epsilon$-cover of $E$. For $\epsilon > 0$, let us define

$$H_\epsilon^s(E) = \inf \sum_{i=1}^{\infty} [\epsilon(U_i]^s \tag{5}$$

where the infimum is over all $\epsilon$-covers $\{U_i\}$ of $E$. The limit of $H^s_\epsilon$ as $\epsilon \to 0$ denoted by $H^s(E)$, is the $s$-dimensional Hausdorff measure of $E$.

Let us notice that in the space $R^l$ one can prove that $H^l(E) = \kappa_l L^l(E)$, where $L^l$ is the $l$-dimensional Lebesgue measure and $\kappa_l$ is a ratio of volume of the $l$-dimensional cube to $l$-dimensional ball inscribed in the cube.

It is evident that $H^s_\epsilon(E)$ increases as the maximal diameter $\epsilon$ of the sets $U_i$ tends to zero, therefore, it requires to take finer and finer details, that might not be apparent in the larger scale into account. On the other hand for the Hausdorff measure the value $H^s(E)$ decreases as $s$ increases, and for large $s$ this value becomes 0. Then the Hausdorff dimension of $E$ is defined by

$$\dim(E) = \inf\{s : H^s(E) = 0\}, \tag{6}$$

and it can be verified that $\dim(E) = \sup\{s : H^s(E) = \infty\}$.

The Hausdorff dimension is one of several fractal dimensions. To make the definitions more evident let us notice that for the Hausdorff dimension of the Peano curve has dimension 2 and of the Cantor middle set is $\log 2/\log 3$, while its topological dimension $D_T$ is zero. In most cases Hausdorff dimension $\geq$ the topological one. In its classical form a fractal is by definition a set for which the Hausdorff dimension strictly exceeds the topological dimension.

Topological dimension takes non-negative integer values and is invariant under homeomorphism, while the Hausdorff dimension is invariant under bi–Lipschitz maps (sometimes called quasi–isometries).

As some approximation of the fractal dimension one may consider another dimension known as the packing dimension or the box-counting dimension [4]. To calculate this dimension for a set $S$ imagine this set lying on an evenly-spaced grid. Let us count how many boxes are required to cover the set. The **box-counting** dimension is calculated by observing how this number changes as we make the grid finer. Suppose that $N(\epsilon)$ is the number of boxes of the side length $\epsilon$ required to cover the set. Then the box-counting dimension is defined as:

$$\dim_{box}(S) = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)} \tag{7}$$

It is possible to define box–counting dimensions using balls, with either the covering number or the packing number. It follows that box dimension of $E$ is always $\geq \dim(E)$.

By inventing the fractal (Hausdorff) dimension the trajectory of GA's or its attractor can be investigated. Algorithms could be regarded as equivalent if they have the same computational complexity while solving the same problem. As the measure of computational complexity of genetic algorithm, we propose a product of population's size and the number of steps after which an optimal solution is reached. This measure of computational complexity of genetic algorithms joins the memory and the temporal complexity.

During the execution of genetic algorithms, a trajectory is realized and should "converge" to some attraction set. It is expected that an ideal genetic algorithm produces an optimal solution which, in the term of its trajectory, leads to an attractor which is one–element set. On the other hand, for an algorithm without selection the attractor is the whole space. Then, we could say that algorithms are equivalent when they produce similar attractors [5].

Hence, instead of the entropy, the fractal dimension will be used as an indicator, or better to say—a measure of the classifications of GAs.

We say that two genetic algorithms are equivalent if they realize trajectories with the same fractal dimension.

## VI. Compression ratio

It is our conjecture that some characteristic feature of the trajectory of GA can be obtained by analysing the ration of the compressed trajectory to itself. We decided to investigate Lempel-Ziv compression algorithm [2] applied to populations executed by various genetic algorithms. We implemented five De Jong's functions with 10 different parameters sets. Each experiment was run 10 times. All together we obtained 500 different trajectories. The following settings of algorithms were considered

| EXP | CROS | PC | PM | SEL |
|---|---|---|---|---|
| 1 | 1 | 0.25 | 0.001 | t |
| 2 | 2 | 0.6 | 0.01 | r |
| 3 | u | 0.95 | 0.05 | p |
| 4 | 1 | 0.6 | 0.05 | p |
| 5 | 2 | 0.25 | 0.001 | r |
| 6 | u | 0.6 | 0.01 | t |
| 7 | 1 | 0.95 | 0.01 | r |
| 8 | 2 | 0.95 | 0.001 | p |
| 9 | u | 0.25 | 0.05 | t |
| 10 | 1 | 0.95 | 0.99 | r |

where EXP is the experiment number; CROS is type of crossover operator (one point, two point, uniform); PC and PM are probabilities of crossover and mutation, respectively; and SEL is type of selection operator (tournament, rank, and proportional). In each experiment the population consisted of 25 points and the genetic algorithm was realized on 100 generations (points).

We performed numerous experiments on compressing particular generations with Lempel-Ziv algorithm of various bit resolution. We measured number of prefixes resulting from compression process and corresponding compression ratio in scenarios of two types. The first one considered single generations, and for each trajectory we obtained corresponding trajectory of number of prefixes used. In the second scenario, each next generation was added to all past generations forming an ascending family of sets of generations. Compressing elements of such family gave an overall picture how number of prefixes used in the compression stabilizes over time.

## VII. Experiments and conclusions

The first experiments with attractors generated by GAs and the expression (8) have been performed by our co-worker in [5]. His results allow us to claim that the present approach can be useful in the GA's dynamics research.

In the recent paper [14] we have used another approach to the approximation. Let $N(T,\epsilon)$ be the minimum number of $K$-dimensional cubes with the edge length equal to $\epsilon$, that covers the trajectory $T \subset X$, and $X$ is a $l$-dimensional search space. To be more evident let us consider the case when $\epsilon = 2^{-k}$ and diminish the length of cube edges by half. Then the following ratio will approximate the box counting dimension of trajectory $T$

$$D_c(T) \approx \frac{\log_2 N(T, 2^{-(k+1)}) - \log_2 N(T, 2^{-k})}{\log_2 2^{k+1} - \log 2^k} = \\ = \log_2 \frac{N(T, 2^{-(k+1)})}{N(T, 2^{-k})}. \qquad (8)$$

The approximated expression (8) of the box dimension counts the increase in the number of cubes when the length of their edges is diminished by half.

In [14] we have included new calculation results. 12 benchmark functions were used (cf. [6]) in the analysis. Experiments were performed for different dimension: 10, 15, 20 bits with operator parameters and Popsize. Then the box counting dimension was used to calculate the trajectory dimension.

As far as the analytical approach and the formal definitions of dimensions (5) and (6) are concerned their computer implementation needs additional investigations. Computer accuracy is finite, hence all limits with $\epsilon$ tending to zero will give unrealistic results. For example, if the calculation drops below the computing accuracy the expression value becomes zero or undefined. It means that we have to stop taking limit values in early stage. Hence, the questions arise: to which minimal value of $\epsilon$ the calculation should be performed and whether and how the relations with limits should be substituted with finite, non-asymptotic, expression? This, however, will be the subject of our further research.

The main idea of our experiments made in [14] was the verification and confrontation of our theoretical considerations and conjectures with real genetic algorithms.The analysis of the experimental result has shown that the value of box-counting dimension of the trajectory of genetic algorithms is not random. When we use the same fitness function and the same configurations, then the box dimensions become clustered near the same value. Whole trials of the independent running attains the same values. Moreover with the different functions but the same configuration we deal with the conservation of box-counting dimension clustering.

Average values of the box-counting dimension for the united trajectories of the algorithms from the same trial were similar to these which were calculated by averaging of the dimension of individual trajectories. This fact acknowledges the conjectures that box-counting dimension could characterize the complexity of algorithms.

Box-counting dimension describes the way of evolution during search. Algorithms which attain the maximum in a wide loose set have bigger dimension than others which trajectories were narrow, with small differences between individuals.

One can say that bigger box dimension characterizes more random algorithms. The main result of the experiments states that fractal dimension is the same in the case when some boxes contains one individual as well as when these boxes contain many elements (individuals). Box dimension does not distinguish the fact that two or more elements are in the same place. They undergo counting as one element. The value of dimension should depend on the number of elements placed in each box. Our main conclusion is that good characterization is the information dimension.

## ACKNOWLEDGEMENT

The research work on the paper done by W.K., S.K., Z.M. and P. S. was supported by the KBN Project No. 3 T11 C007 28 in 2005-2007. The authors are grateful to Dr. J. Socała for valuable discussions.

## REFERENCES

[1] Barnsley M. F.: Lecture notes on iterated function systems, in *Chaos and Fractals. The Mathematics Behind the Computer Graphics, Proc. Symp. Appl. Math.*, vol. 39, R. L. Devaney and L. Keen (eds.) American Mathematical Society, Providence, Rhode Island, pp. 127–144, 1989.

[2] G. Frizelle G., Suhov Y. M.: An entropic measurement of queueing behaviour in a class of manufacturing operations. *Proc. Royal Soc. London A* (2001) **457**, 1579–1601.

[3] Friedman N. A., Ornstein D. S.: On isomorphisms of weak Bernoulli transformations, *Adv. in Math.*, **5**, 365-394, 1970.

[4] Harrison J.: An introduction to fractals, in *Chaos and Fractals. The Mathematics Behind the Computer Graphics, Proc.Symp. Appl. Math.*, vol. 39, R. L. Devaney and L. Keen (eds.) American Mathematical Society, Providence, Rhode Island, pp. 107–126, 1989.

[5] Kieś P.: Dimension of attractors generated by a genetic algorithm, in *Proc. of Workshop Intelligent Information Systems IX held in Bystra, Poland, June 12–16*, IPI PAN, Warszawa, pp. 40–45, 2000.

[6] Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd, rev. edition, Springer, Berlin, Heidelberg, 1996.

[7] Ossowski A.: Statistical and topological dynamics of evolutionary algorithms, in *Proc. of Workshop Intelligent Information Systems IX held in Bystra, Poland, June 12-16*, IPI PAN, Warszawa, pp. 94–103, 2000.

[8] Ornstein D. S.: *Ergodic theory, Randomness and Dynamical Systems*, Yale Univ. Press, 1974.

[9] Vose M. D.: Modelling Simple Genetic Algorithms, *Evolutionary Computation*, **3** (4), 453–472, 1996.

[10] Wolpert D.H. and Macready W.G.: No Free Lunch Theorems for Optimization, *IEEE Transaction on Evolutionary Computation*, **1** (1), 67-82, 1997, http://ic.arc.nasa.gov/people/dhw/papers/78.pdf

[11] Igel, C., and Toussaint, M.: A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions, *Journal of Mathematical Modelling and Algorithms*, **3**, 313–322, 2004.

[12] English, T.: No More Lunch: Analysis of Sequential Search, *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pp. 227–234. 2004, http://BoundedTheoretics.com/CEC04.pdf

[13] Szlenk W., *An Introduction to the Theory of Smooth Dynamical Systems.*, PWN, Warszawa,John Wiley& Sons, Chichester, 1984 G. H.

[14] Kotowski S., Kosiński W., Michalewicz Z., Nowicki J., Przepiórkiewicz B., Fractal dimension of trajectory as invariant of genetic algorithms *ICAICS, 9-th International Conference on Artifical Intelligence and Soft Computing, 2008*, LNAI 5097, Springer, Berlin, Heidelberg, New York, pp. 414–425.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Evolution of Strategy Selection in Games by Means of Natural Selection

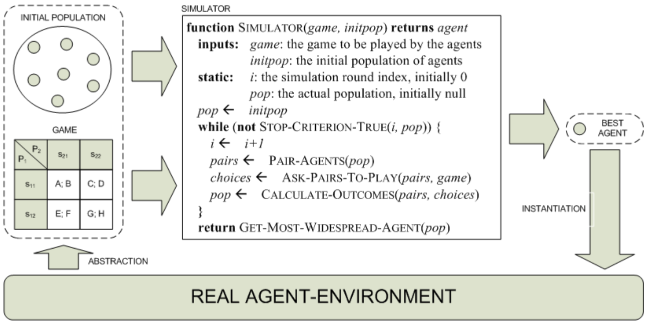
Daniel L. Kovacs, Tadeusz Dobrowiecki
Budapest University of Technology and Economics, H-1117 Budapest, Hungary
Email: {dkovacs, dobrowiecki}@mit.bme.hu

*Abstract*—**In this paper we present a novel agent-based simulation model for the natural selection of replicating agents whose survival depends on their programs for selecting their strategy to interact with each other. Game theoretic models describe this kind of interaction. The simulator can be used both for analysis, design and verification of autonomous systems (by intuitively abstracting them into our model and running the simulation). Good system design can be selected even for difficult, underspecified design problems. Although the inspiration of the model comes from evolutionary game theory, the evolving agents may represent not only biological, but also technical systems (e.g. software agents), and thus in some cases the underlying evolutionary mechanisms and observations differ from the typically published results.**

## I. Introduction

THIS article presents a novel, realistic agent-based simulation model for the natural selection of replicating agents whose survival depends on their programs for selecting their strategy to interact with each other [1]. Compact game theoretic models describe such interaction, masking most low-level details, but nevertheless catching the essential program features, and simplifying simulation [2].

The implementation of the proposed simulation model is a domain-independent problem-solving tool that facilitates not only analysis, but also design and verification of autonomous systems. Assuming that there is a fixed set of different *design alternatives* (e.g. different types of software agents) in the real environment which interact with each other repeatedly and randomly in a pairwise manner, and that their number depends on their success in these interactions, they can be modeled – including their environment – by **(1$^{st}$)** a fixed pool of game playing agents (being able to replicate and terminate) and **(2$^{nd}$)** a 2-person game. These are the two inputs of our simulation model as shown in Fig. 1.



| P₁ \ P₂ | s₂₁ | s₂₂ |
|---|---|---|
| s₁₁ | A; B | C; D |
| s₁₂ | E; F | G; H |

```
function SIMULATOR(game, initpop) returns agent
  inputs:  game: the game to be played by the agents
           initpop: the initial population of agents
  static:  i: the simulation round index, initially 0
           pop: the actual population, initially null
  pop ←  initpop
  while (not STOP-CRITERION-TRUE(i, pop)) {
    i ←  i+1
    pairs ←  PAIR-AGENTS(pop)
    choices ←  ASK-PAIRS-TO-PLAY(pairs, game)
    pop ←  CALCULATE-OUTCOMES(pairs, choices)
  }
  return GET-MOST-WIDESPREAD-AGENT(pop)
```

Fig. 1 The simulation model and its use in system analysis and design

The simulation is boldly as follows: agents are repeatedly and randomly paired to play the 2-person game, and their survival (and chance for replication) depends on the utility they can gather during that process. Thus quantitative aspects enter the simulation via the proportion of subpopulations of these different agents constituting the initial population. Evolution essentially means the dynamics of the distribution of these different agents (species) within the evolving population. This way we intend to find the agent corresponding to the best design alternative even when the under-defined specification (usually due to missing, or uncertain knowledge about the task environment) does not make it possible to pinpoint the optimal design via the traditional design process. The success of this process depends very much on the abstracting phase, where the real problem is represented as an input for our simulation model.

The inspiration for the model comes mainly from evolutionary game theory, and the seminal experiments of Robert Axelrod with the **T**it-**F**or-**T**at (**TFT**) strategy in the repeated **P**risoner's **D**ilemma (**PD**) [3]–[5]. Several other models of program evolution exist, but our approach differs from them mainly in the following [6]–[8].

1. We utilize natural selection as a realistic "driving force", and not some preprogrammed, explicit mechanism, to evolve the population;
2. we do not consider the emergence of new variants, only proliferation, i.e. asexual replication (spreading);
3. we do not impose any formal constraints on the structure and inner workings of the agent programs.

We hope thus to model, predict, and support design in several interesting real world situations (e.g. the success of given software agents managing resources or being engaged in electronic commerce on the Internet).

The rest of the paper is organized as follows: Section 2 introduces the background of the simulator, Section 3 describes its architecture and implementation in detail, Section 4 presents and evaluates some essential experiments, Section 5 contains conclusions and outlines future research.

## II. Background

In the following, we will briefly summarize those approaches, which mainly influenced our model. The purpose of this is to introduce some fundamental concepts, and to enable later discussion of similarities and differences.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

### *A. Axelrod's experiments*

The goal of Axelrod's experiments was to find a program which efficiently plays the repeated PD game (cf. Table I). At first 15, then 62 programs were chosen, and pairwise compared. Every program played against each other a fixed number of rounds. In every round they had to choose between two strategies (cooperate or defect), and got their respective payoff according to their collective choice.

TFT, a simple program, which initially cooperates, and then repeats the previous choice of its opponent, won both tournaments by collecting the most at the end. Axelrod concluded, that because of the importance of PD as a model of social interaction, the core characteristics of cooperation in general must be those found in the TFT.

Axelrod then conducted other experiments to confirm the success of TFT, called ecological and evolutionary analysis. In the former he examined the dynamics of a population composed of programs submitted for the second tournament (all having an equal proportion of the initial population). The proportion of a program changed according to the proportion-weighted average of its scores against other programs in the second tournament. The results of this experiment again underpinned the success of TFT. Its proportion was the largest, and it grew steadily.

In the latter experiment Axelrod used genetic algorithms to evolve programs (represented as bit strings) that could play against the programs submitted for the second tournament [9]. Fitness was equal to the then average payoff. The algorithm produced programs that played effectively against the tournament programs, and resembled TFT.

### *B. Evolutionary Game Theory*

Evolutionary game theory, on contrary to the previous simulations, enables formal analysis and prediction of such evolving systems (although only for relatively simple cases).

For example, let's suppose that we have an infinite population of agents, who strive for resources. The game is divided into rounds, and in every round every agent randomly (according to uniform distribution) meets an other agent to decide upon a resource of value $V>0$ . For the sake of simplicity let's say, that there are only two types of agents: hawks (aggressive), and doves (peaceful).

When two hawks meet, they fight, which costs $C$ , and so they get $(V-C)/2$ per head. When two doves meet, they divide the resource equally between each other (they get $V/2$ per head). When a hawk meets a dove, then the hawk takes the resource (gets $V$ ), while the dove is plundered (gets $0$ ).

Table I.
Payoff matrix of the "Prisoner's Dilemma" game used by Axelrod

| Player 2<br>Player 1 | Defect | Cooperate |
|---|---|---|
| Defect | 1; 1 | 5; 0 |
| Cooperate | 0; 5 | 3; 3 |

This situation is simply modeled by the **H** awk- **D** ove (**HD**) game (cf. Table II) [10]. The gained payoffs are collected over rounds, and the proportion of hawks and doves depends on their average collected payoff. Formally this is as follows.

Let $p_H^i$ and $p_D^i$ denote the proportion, while $W_H^i$ and $W_D^i$ the average collected payoff of hawks and doves in round $i$ respectively. For every $i \geq 0$ $p_H^i, p_D^i \in [0,1]$ , and $p_H^i + p_D^i = 1$ is true. The average collected payoff of the whole population in round $i$ , $W^i$ , is

$$W^i = p^i_H W^i_H + p^i_D W^i_D \tag{1}$$

The proportion of hawks and doves in round $i+1$ is calculated according to discrete replicator dynamics.

$$p_H^{i+1} = p_H^i \frac{W_H^i}{W^i} \text{ , and } p_D^{i+1} = p_D^i \frac{W_D^i}{W^i} \tag{2}$$

The average collected payoff of hawks and doves in round $i+1$ is respectively.

$$W_H^{i+1} = W_H^i \left( p_H^{i+1} (V-C)/2 + p_D^{i+1} V \right) \tag{3}$$

$$W_D^{i+1} = W_D^i + p_D^{i+1} V/2 \tag{4}$$

Consequently, for example, it can be simply shown that if $V > C \geq 0$ , $W_H^0 = W_D^0 > 0$ , and $p_H^0 > 0$ , then

$$\lim_{i \to \infty} p_H^i = 1 \text{ , and } \lim_{i \to \infty} p_D^i = 0 \tag{5}$$

This means, that the system converges to a state, where only hawks remain in the population. Many similar, interesting results can be obtained this way, although the necessary assumptions are usually unrealistic, and overly simplified (infinite number of agents; trivial programs, that can be handled analytically; etc) [11]. For more realistic and complex cases, with arbitrary programs (like in the ecological analysis of Axelrod) and finite, overlapping generations of varying size, we need to use simulations.

## III. Simulation Driven by Natural Selection

The proposed simulation model combines the advantages of the previous approaches without their drawbacks. It resembles artificial life in many aspects, but it is different in its purpose (it tries to capture the key features of not only biological, but also technical systems' evolution) [12]. It is an extension to the previous approaches, differing from them in the following.

Table II.
Payoff matrix of the "Hawk-Dove" game

| Player 2<br>Player 1 | Hawk | Dove |
|---|---|---|
| Hawk | (V-C)/2; (V-C)/2 | V; 0 |
| Dove | 0; V | V/2; V/2 |

1. Populations are finite, and vary in size;
2. Generations are overlapping;
3. Agents are modeled individually;
4. The selection mechanism, and the fitness of agents is not explicitly given, but it is a product of agents' features and their interaction;

5. Modeling of not only biological, but also technical systems' evolution is considered.

These differences make the model more realistic, and enable us to use it not only for analysis, but also for design.

### C. Detailed Description of the Simulation Model

The basis of the model is an intuitive combination and extension of the ideas discussed in Section 2. The simulation is divided into rounds. There is a finite number of agents in the population, who are randomly paired in every round (according to uniform distribution) to play a 2-person game in the role of one of the players. Every agent of the population plays the same type of game (e.g. just PD, or just HD) in every round of a run, and each of them has a program (e.g. TFT, Random, Always-Cooperate, Always-Defect) for selecting its strategy in these plays.

After a pair of agents finished to play in a given round, the respective (possibly negative) payoffs are added to their individually cumulated utility. If the utility of an agent gets below a level (e.g. zero), then the agent dies, i.e. it instantly disappears from the population, and won't play in the following rounds; otherwise it may reproduce depending on its reproduction strategy. Every agent has a reproduction strategy (defining how, and when to reproduce), but only asexual proliferation, i.e. replication without change is considered. After every agent finished the given round (died, survived, or even replicated), comes the next round.

It can be seen, that in this model there is no explicit selection mechanism directly controlling the proportion of agents in the population (like replicator dynamics, or roulette wheel), but it is an implied, emergent phenomena. Every agent has its own lifecycle (they are born with given features, interact with each other, change, maybe even reproduce, and then possibly die), and only those are "selected" for reproduction, whose features make them able to survive in the environment up to that moment. In our view this process is real *natural selection*.

Of course there are many other more or less similar definitions in literature originating mostly from Darwin [13]. For instance, it is usual to say that "natural selection is a process that affects the frequency distributions of heritable traits of a population" [14] (page 16), and that "heritable traits that increase the fitness of their bearers increase in frequency in a population" [15] (page 821), etc. These statements do not contradict our views, but they aren't specific enough in the sense, that they allow even the selection behind a genetic algorithm to be called "natural". The reason for that is that these statements stem from biology and genetics, which are concerned with natural systems [16], [17]. This is why we use a "new" definition.

Moreover, we need to differentiate between modeling natural selection of natural, and technical systems. In respect of technical systems, "natural selection" may reflect the "natural" peculiarities of technical application areas, which may be quite far from what can be considered natural in biological, or social science. We differentiate these aspects by introducing agents' *reproduction strategies*.

We'll consider two types of reproduction: type 1 is called "natural", and type 2 is called "technical". Agents with type 1 reproduction can have only a limited number of offsprings in their lifetime (maximum one per round). They replicate, if their utility exceeds a given limit (limit of replication). After replication, their utility is decreased with the cost of replication (which is usually equal to the limit of replication). Offsprings start with zero utility, and the same program, and features, as their parents originally (i.e. the same lower limit of utility necessary for survival, limit and cost of replication, and limit on the number of offsprings).

On the other hand, agents with type 2 reproduction can have unlimited offsprings (maximum one per round). They replicate when their utility exceeds the limit of replication, but this limit is doubled every time an offspring is produced, and their utility does not decrease. Offsprings start with the same utility, program, and features, as their parents at the moment of replication (i.e. the same lower limit of utility necessary for survival, and limit of replication).

The rationale behind the above design choices originates in the following: biological agents can reproduce only a limited number of times during their lifetime, while technical systems (e.g. software agents) usually do not have such limitations. The replication (copy) of a technical system is usually inexpensive compared to the replication of a biological agent. Offsprings of technical systems may be exact copies of their parents with the same level of quality, while offsprings of a biological system usually develop from a lower (immature) level. The level of "maturity", where a biological system can reproduce, may be constant, and the same as its parents', but it may increase in the case of a technical system (to represent the increasing cost of production, or the amount of resources needed, etc).

### D. Using the Simulation Model for Analysis and Design

As mentioned before, the simulation model can be used not only for description, but also for design of technical systems. Let's suppose, for example, that we want to design an efficient software agent (e.g. a broker agent, or a trading agent on the Internet acting on behalf of human users, trying to maximize their profit). Our goal is to make this agent as efficient and useful as possible in hope that in result it will spread better (e.g. more users will start to use it). Of course there are many other factors (like marketing, ergonomics, etc) that may influence the success of such a system, but, as for a Designer, ensuring functional efficiency and usefulness is a primary objective. Nonetheless these properties depend not only on the program of the agent, but also on its environment (e.g. the other concurrent agents), which may be not fully accessible, too complex and dynamic to be modeled and analyzed exactly in order to calculate an optimal system design (e.g. in the case of Internet).

It is inevitable to abstract the problem. We recommend game theory to model the strategic interaction in the real agent-environment. It may mask most of the subtlety of agents' programs, but it catches essential features, and works toward a simpler simulation, which becomes necessary, if – according to the conclusions of Section 2 – agents' programs are too complex to be handled analytically.

For example, let's suppose that software agents participate in electronic auctions on the Internet to acquire goods for their human users. The less they pay for them, the higher their individual utility is (from the perspective of their users). If two agents agree to cooperate (e.g. to raise the bid only until one of them wins and then share equally), then they pay less, than being non-cooperative (e.g. when competing against each other by bidding more and more to win). The best outcome is however for an agent to "cheat" its cooperative partner by defecting (e.g. raising the bid to win while the other is not competing and then not sharing the win). This outcome is the worst for the other agent since it gets nothing. Let's say, that if a software agent is successful, then after it achieves a given profit for its user, then its user invests in another such agent, but if the profit is below a level then she/he stops using it. *What software agent would we design to better survive and proliferate in this scenario?*

The situation can be modeled by the proposed simulation: agents have two strategies (cooperate or defect), and the order of their preferences corresponds to a PD game (cf. Section 4/A). Now we can estimate the distribution of the various software agents in the real agent-environment, and construct an initial population accordingly. Assuming that natural selection is the "driving force" behind agents' evolution, we can give the corresponding PD game and the initial population as an input to the simulation model, and run it. If we run the simulation by injecting several different candidate solutions into that initial population, we can get the most successful variant. This way we can use natural selection to support not only analysis, but also design.

### *E. Some Thoughts About Natural Selection*

But why do we need natural selection? Why don't we use some other, explicit model (e.g. replicator dynamics) for simulating the dynamics of the real environment? – First, we claim that using natural selection instead of some explicitly declared mechanism for evolving the population of agents allows more realistic predictions of several interesting real world phenomena. In Section 4 we'll discuss this in more detail (in relation to experiments concerning replicator dynamics). Second, if the selection mechanism is direct and explicit, then it usually accounts for some kind of fitness to measure the goodness of individuals. Such a value would reflect the goodness of all (maybe even hidden or implicit) features influencing the success of an agent in its environment. The calculation of such a fitness value can be hopeless for complex, dynamic systems and environments.

In the proposed simulation we currently use a cumulated utility value associated with the goodness of an agent. But it is not a fitness value. The fitness of an unvarying individual can change only when its environment changes. But this is not true for the cumulated utility, which is only a parameter of the agent's state, and may change even when the agent, its program, or its environment does not vary. Also, the survival of an agent may depend on much more sophisticated terms than just checking if the cumulated utility value is below a lower limit. This mechanism could be easily exchanged by a much more complex and realistic variant.

### *F. Implementation Issues*

The proposed simulation model was implemented in the **JADE** (**J**ava **A**gent **DE**velopment) framework [18]. It is an open-source, Java-based, platform independent, distributed middle-ware and **API** (**A**pplication **P**rogramming **I**nterface) complying with the **FIPA** (**F**oundation for **I**ntelligent **P**hysical **A**gents) standards [19]. It enables relatively fast and easy implementation of physically distributed, asynchronous, high level multi-agent systems.

The implemented software architecture was aimed to be as simple, as possible. It consisted of only two JADE agents: a `GameAgent` (`GAg`), and a `PlayerAgent` (`PAg`). `GAg` was responsible for conducting the runs of the simulation, and orchestrating `PAg`-s, while `PAg`-s were the actual agents in the population, who were paired in each round to play a given 2-person game.

Each JADE agent had a variety of (mostly optional) startup parameters, which in case of a `GAg` set the type of the game to be played (e.g. PD, or HD, or else), the maximal number of agents in the population, and the maximal number of rounds in the run. The OR-relation of the latter two defined the termination criteria of a run. The startup parameters of a `PAg` set agents' program and reproduction strategy, initial utility, the lower limit of utility, the limit and cost of reproduction, the limit on the number of offsprings, and the capacity of memory. The latter was needed because each agent had to be able to use its percept history in order to decide upon the strategy to be played in a given round. The percept history of an agent associated a series of events (information about past plays) to agent identifiers (ID-s). There was a limit on maximal length of these series and maximal number of ID-s. If any of these limits was exceeded then the oldest element was replaced by the new one.

Now the simulation went as follows. First a given number of `PAg`-s were started on the JADE agent platform (constituting the initial population), followed by a `GAg`, who at the beginning of every round first searched the platform for available `PAg`-s (because later there may have been newly born agents, or some agents terminated), then made a pairing of the `PAg`-s, and informed these pairs about the game to be played (who plays with whom and in what role). The pairs of `PAg`-s then replied to the `GAg` with a chosen strategy ID respectively. The `GAg` then calculated their respective payoff accordingly and informed them about it. This was repeated until the termination criterion was satisfied. Several interesting experiments were conducted this way. Some of them are explained in the next section.

## IV. Experiments

The experiments consisted of running the simulation described above with several different initial populations and games to observe the changes in the number, proportion, and average utility of the different types of agent programs. Since JADE is a distributed framework (and for faster performance) a cluster of 10-13 PC-s (with 1-2 Ghz CPU, 0,5-1 Gb RAM, Win. XP, and JADE 3.5) was used to run the simulations.

Each experiment had its own settings, but a part of them was the same for every experiment. The maximal number of agents was 800; the maximal number of rounds was 250; the maximal number of offsprings was 3; the limit and the cost of reproduction was 20; the lower limit of agents' utility and their initial utility was 0; the maximal number of percept histories (about different opponents) was 1000; and the limit on the length of such a percept history was 4 for every agent in every experiment. Everyone was playing in every round (except when the number of agents was odd).

In the following we will describe these experiments grouped according to the games the agents' were playing. Five games were examined: **P**risoner's **D**ilemma (**PD**), **C**hicken **G**ame (**CG**), **B**attle of **S**exes (**BS**), **L**eader **G**ame (**LG**), and **M**atching **P**ennies (**MP**). The first 4 games are the only interesting (i.e. dilemmas, where the choice is not trivial) between the 12 symmetric out of altogether 78 ordinally different 2×2 games [20]. MP is an addition to these because unlike them it has no pure strategy **N**ash **E**quilibrium (**NE**), and this has some interesting implications (cf. Section 4/E) [21].

During the experiments agent programs were drawn from a fixed set. Only the following programs were studied yet: Always-Strategy1, Always-Strategy2, TFT, and Random. Nonetheless both types of agents' reproduction strategy (cf. Section 3/A) were examined. All in all, this configuration was more than enough to run insightful experiments comparable to the previous results discussed in Section 2.

### *G. Prisoner's Dilemma*

PD is one of the most popular 2-person games in game theory [22]. It is a special case of the HD game, when $V > C \geq 0$ (cf. Table II). The original story of the game is about two prisoners, who are put in separate cells (cannot communicate), and are asked to simultaneously decide, whether to cooperate, or defect. The best outcome is defecting, if the other cooperates, but it is the worst outcome for the other. It is better if both defect, and even better, if both cooperate. The game is called a "dilemma" because its only NE is the sub-optimal Defect-Defect outcome.

Similar situations may arise in the world of artificial systems too (cf. example in Section 3/B). If the payoffs are chosen according to the HD game, where V=4, C=2 (and so it becomes a PD), then if the initial population consists of altogether 6 agents: 3 Always-Cooperate, and 3 Always-Defect agents, then the proportions of the different agent programs change according to Fig. 2, which is in accordance with the formal predictions of Section 2/B (cf. Eq. 5). Defective agents (hawks) infest the population, and the proportion of cooperative players (doves) steadily converges to zero. The reproduction strategy of agents in Fig. 2 is of type 1 ("natural"), but essentially the results are the same in case of type 2 ("technical") reproduction.

Fig. 3-5 show the change of the quantity, and average utility, if the initial population consists of 3 Random, and 3 TFT agents. The quantity of these subpopulations ("species") does not decrease because there are no negative payoffs in the game, and so agents cannot die since their cumulated utility cannot decrease below the lower limit.

Fig. 5 shows the same situation as Fig. 4 except that the reproduction is "technical". A periodical change of the average utility of subpopulations can be observed on Fig. 4, while it is monotonously increasing on Fig. 5. That is because "natural" reproduction has a cost (equal to the limit of reproduction) on the contrary to "technical" reproduction.

According to Fig. 3, Random agents typically outperform TFT agents by far. This is the case with both types of reproduction. Similarly, Always-Defect agents also outperform TFT agents. These observations seem to differ from the results mentioned in Section 2/A, although the latter is not so surprising, since the numerical values of the game are now such, that the defective player gets as much against a cooperating player, as two cooperating players together, and its average income (5/2) is better than by cooperating (2/2).

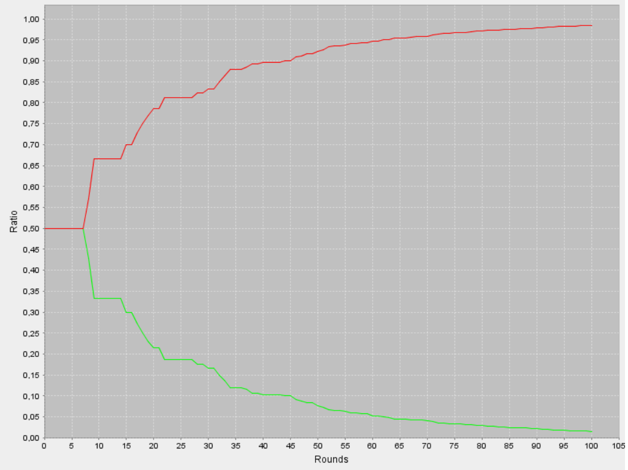


Fig. 2 Typical change of agent programs' proportion in PD, if the initial population consists of 3 Always-Cooperate (green), and 3 Always-Defect (red) agents whose reproduction strategy is "natural".

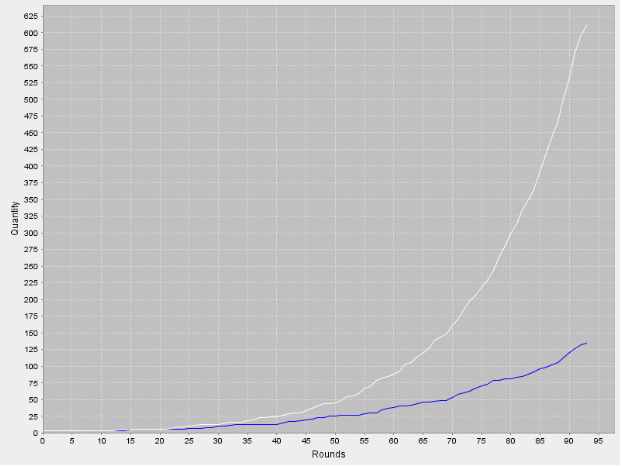


Fig. 3 Typical change of agent programs' quantity in PD, if the initial population consists of 3 Random (white), and 3 TFT (blue) agents whose reproduction strategy is "natural".

Moreover, according to Fig. 3-5, the change of subpopulations' proportion isn't in direct proportionality with the ratio of their average utility and the average utility of the whole population, as predicted by replicator dynamics (cf. Eq. 2). It must be pinned down, that replicator dynamics holds a central role among models of evolutionary dynamics. Some consider it even to be a fundamental theorem of natural selection [23]-[25]. Reference [26] even writes "The dynamics of such replicator systems are described by the fundamental theorem of natural selection". Nonetheless, according to these experiments, replicator dynamics seems to be unable to model such scenarios properly. Thus, in our view, the assumptions behind Eq. 2 are unrealistic.

If, for example, there are two subpopulations, and one of them grows exponentially (which is common in evolution), then after a while their average utility will determine the average utility of the whole population, and so the ratio of these two averages will converge to 1. In this case replicator dynamics would predict that the growth of the large subpopulation stops (cf. Eq. 2). But why should it?

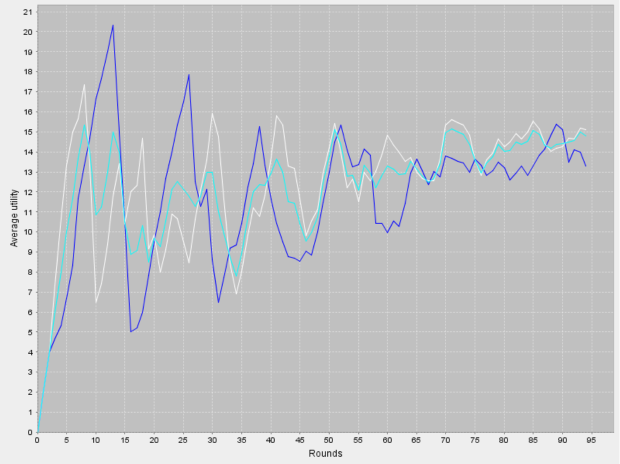


Fig. 4 Typical change of agent programs' average utility in PD, if the initial population consists of 3 Random (white), and 3 TFT (blue) agents whose reproduction strategy is "natural". The average utility of the whole population is also shown (light blue).

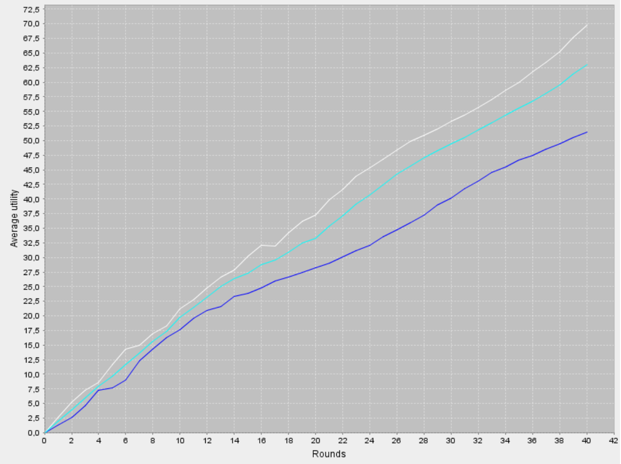


Fig. 5 Typical change of agent programs' average utility in PD, if the initial population consists of 3 Random (white), and 3 TFT (blue) agents whose reproduction strategy is "technical". The average utility of the whole population is also shown (light blue).

The larger the population, the more offsprings are born. The tendency of growth accelerates, if it is not limited by finite resources, or some other way. So maybe there is an implicit assumption (e.g. about finite resources) in Eq. 2. Apparently after a while every evolutionary system should reach its boundaries, but until then their growth must be governed by a dynamics different from replicator dynamics.

The results shown in Fig. 6 utter these concerns further. Axelrod's ecological analysis (cf. Section 2/A), based on replicator dynamics, predicts the extinction of the invading defectors, but our experiments show the opposite is true [7].

All of the above results are typical in a sense that they are indifferent to parameter changes (e.g. changing the size of the initial population; changing the type, the limits, or the cost of reproduction, or the payoff values of the game). TFT agents could be made a little better by increasing their memory, but in the end it doesn't change the overall tendency. These observations may also help to explain the scarce evidence of TFT-like cooperation in nature [27].

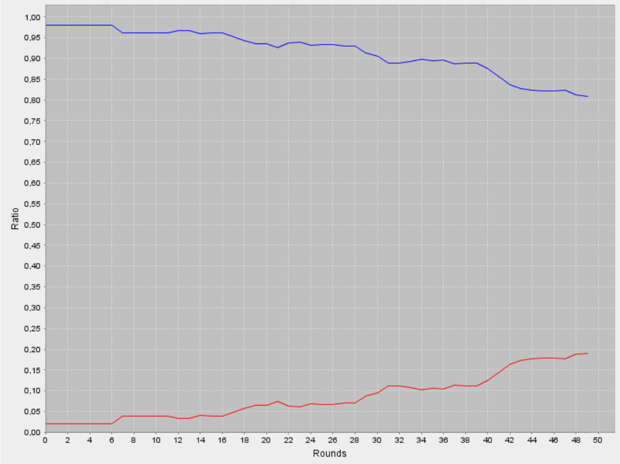


Fig. 6 Typical change of agent programs' proportion in PD, if the initial population consists of 50 TFT (blue), and 1 invading Always-Defect (red) agent(s) whose reproduction strategy is "natural".

### *H. Chicken game*

This game is also a special case of the HD game, when $C > V$ (cf. Table II) [22]. The original story is about two cars driving toward each other. If both drivers are "reckless", and won't swerve away, it is the worst outcome, since they crash. Better is, if one swerves away (being "chicken"), while the other wins (best outcome). But it is better for the former, if the latter swerves away too. This is a mixed motive game, since it has two NE-s (if players do the opposite).

A technical scenario may be the same as in the previous example (cf. Section IV/A) with the exception that if both agents are non-cooperative (reckless), then it is the worst outcome for them (e.g. because if two such cheating bidder agents meet, they disclose each other, and thus are excluded from the auction, and must pay a penalty to the organizers).

Experiments with this game showed different results than in case of PD in Section 4/A. The main reason for that is that the payoffs were chosen according to HD game, where V=2, C=4, and so agents could die because of negative payoffs.

Always-Strategy2 (i.e. Always-Chicken) proved to be the best (most proliferating) program out the four studied alternatives if there were only two types of programs in the initial population. Always-Reckless and TFT claimed the second place, while Random was the worst. This means that if the cost of being mutually defective is beyond the achievable value, then it becomes too risky not to cooperate.

It was interesting to observe, that if Always-Reckless proved to be the winner of a situation (i.e. if it extinguished an other "species"), then it too died out. In this aspect Always-Reckless is "parasite", that exploits the other subpopulations from whom its survival depends.

Experiments with more than two types of agent programs were rather unpredictable. They depended mostly on the actual pairing of the individuals in the first dozen of rounds.

### *I. Battle of Sexes*

This game is also a mixed motive game, like CG, but it differs from it in that it is asymmetric by default (cf. Table III) [2]. The original story is about a husband and wife, who must choose between going to a football match, or an opera. The husband would better like to go to the football match, while his wife would better go the opera. But, in any case, it is more preferable for them to be together, than to be alone.

Cooperation is different in the case of the husband, than in the case of the wife. They cooperate, if they try to do what is best for the other, and if they both do that, it is the worst (husband goes to opera, and wife goes to football).

TABLE III.
PAYOFF MATRIX OF THE "BATTLE OF SEXES" GAME

| Wife<br>Husband | Opera | Football |
|---|---|---|
| Opera | 1, 2 | -1, -1 |
| Football | 0, 0 | 2, 1 |

This means, that Always-Cooperate, and Always-Defect strategies are a bit more complex now, since they depend on the role of the agents too. TFT needs also to be revised.

A corresponding technical scenario could be, when two tasks share the same resource (e.g. CPU, or server). Their user wants them to finish as soon as possible. The worst case is, if they don't access the resource at all (e.g. its availability may have a default cost to the user). If they access it at the same time (both defect), it is better, but it slows their execution too much. So it is better for them to access the resource separately, but the first to access it is the best.

Experiments showed that regardless of the type of reproduction, Always-Cooperate and TFT agents were the worst (others made them die out every time). Always-Defect was the best program, and Random was second (it survived).

### *J. Leader game*

This game is similar to the symmetrical form of BS, with the exception that mutual defection is the worst outcome, and mutual cooperation is better [28].

The name of the game comes from the following situation: two cars wait to enter a one-way street. The worst case is, if they go simultaneously, because they crash. If both wait (cooperate), it is better. But it is even better if they go separately. The one, who goes first, is the best (cf. Table IV) .

A technical scenario could be similar to the previous example in Section IV/C with the exception that if both tasks access the resource at the same time, then it breaks (e.g. the server crashes). So this is the worst outcome. It is better, if they do not access it at all, and even better, if they access it separately. The first agent to access the resource is the best.

According to our experiments, TFT and Always-Cooperate were better, than Always-Defect agents, but Random agents again outperformed TFT agents. The reproduction strategy made a difference in the tendencies, but not in the outcome.

TABLE IV.
PAYOFF MATRIX OF THE "LEADER" GAME

| Player 2<br>Player 1 | Go | Wait |
|---|---|---|
| Go | -1, -1 | 2, 1 |
| Wait | 1, 2 | 0, 0 |

### *K. Matching Pennies*

This game is asymmetric (like BS), with the exception that it has no symmetric form, and cooperation and defection have no meaning in it [29]. Thus the first (hitherto cooperative) choice of TFT doesn't particularly matter now.

The original game is about two players, who both have a penny. They turn the penny secretly to heads or tails, and then reveal their choice simultaneously. If the pennies match, one player gets a dollar from the other, else it is conversely (cf. Table V).

TABLE V.
PAYOFF MATRIX OF THE "MATCHING PENNIES" GAME

| Player 2<br>Player 1 | Heads | Tails |
|---|---|---|
| Heads | 1, -1 | -1, 1 |
| Tails | -1, 1 | 1, -1 |

A corresponding technical scenario can be where two software agents compete for resources (e.g. locks to files), and they have two different strategies to get the resources. One of the agents is faster than the other, so if both choose the same strategy, then the faster agent gets all of the resources. Else the slower agent can also get some of them.

Our experiments showed that in this scenario Random agents were the fittest for survival (playing the only mixed NE of the game), but in case of type 1 reproduction they died out like all the others. However in case of type 2 (technical) reproduction they could cumulate enough utility to ensure their survival, and start proliferating after a while.

## V. CONCLUSIONS

In this article we presented a novel agent-based simulation model for natural selection of programs selecting strategies in 2-person games, and its use in system analysis and design. Our goal was (1) to create a framework enabling agent design for complex, underspecified environments (e.g. Internet); (2) to give a realistic model for the natural selection of not only natural, but also technical systems; and (3) to reproduce some decisive experiments.

The framework facilitates agent design by supporting the choice of agents (from a finite set of alternatives). The distinction between natural and technical systems is made by introducing reproduction strategies of the agents. Natural selection is not explicitly present, but emerges from the inner workings of the agent population. Experiments revealed several interesting aspects of such population dynamics. For example, the assumptions behind replicator dynamics were shown to be inherently unrealistic and simplifying; the fitness of the Tit-for-Tat strategy was the opposite of what was expected according to the previous experiments of Axelrod; not just the Prisoner's Dilemma, but several other fundamental 2-person games were examined, and for every such game a concrete technical analogy was also given.

There are many possible ways to continue this research. For instance the simulation can be further refined by allowing agents to play not only one, but several different games during a run. We currently use only 2-person games for modeling strategic interaction between agents. This can be extended to N-person games. More complex programs could be examined in the experiments. We could introduce sexual reproduction instead of the current asexual replication. A genetic representation of agent programs could be given to enable the birth of new variations. Finally, the limits on the cardinality of the agent population could implicitly depend on environmental resource limitations.

## References

[1] J. Haldane, "The theory of natural selection today". *Nature,* 183 (4663), pp. 710–713, 1959.

[2] D. Fudenberg and J. Tirole, *Game theory*, MIT Press, 1991.

[3] J. Maynard-Smith, *Evolution and the Theory of Games*, Cambridge University Press, 1982.

[4] R. Axelrod, *The Evolution of Cooperation*, Basic Books, 1984.

[5] R. Axelrod, *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*, Princeton University Press, 1997.

[6] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* . MIT Press, 1992.

[7] C. Ferreira, "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems," *Complex Systems* , vol. 13, no. 2, pp. 87-129, 2001.

[8] D. L. Kovacs, "Evolution of Intelligent Agents: A new approach to automatic plan design", In: Proceedings of IFAC Workshop on Control Applications of Optimization, Elsevier, 2003.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, 1975.

[10] J. Maynard Smith, "Theory of games and the evolution of animal contests," *Journal of Theoretical Biology* , vol. 47, pp. 209-221, 1974.

[11] J. Hofbauer and K. Sigmund, "Evolutionary game dynamics," *Bulletin of the American Mathematical Society*, vol. 40, pp. 479–519, 2003.

[12] M. Bedau, "Artificial life: organization, adaptation and complexity from the bottom up," *TRENDS in Cognitive Sciences*, vol. 7, no.11, 2003.

[13] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* , John Murray , 1859.

[14] J. A. Endler, *Natural Selection in the Wild,* Monographs in Population Biology 21, Princeton University Press, 1986.

[15] P. Kitcher, "Philosophy of Biology," In: *The Oxford Handbook of Contemporary Philosophy* , ed. Frank Jackson and Michael Smith, Oxford University Press, pp. 819-847, 2008.

[16] D. J. Futuyma, *Evolutionary Biology*, Sinauer Associates Inc., 1998.

[17] D. J. Futuyma , *Evolution* . Sinauer Associates Inc., 2005.

[18] F. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*, Wiley Series in Agent Technology, 2007.

[19] Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org

[20] A. Rapoport, M. Guyer, and D. Gordon *The 2X2 Game* , University of Michigan Press, 1976.

[21] J. F. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[22] W. Poundstone, *Prisoner's Dilemma: John Von Neumann, Game Theory, and the Puzzle of the Bomb*, Anchor Books, 1992.

[23] R. A. Fisher, *The Genetical Theory of Natural Selection*, Clarendon Press, Oxford, 1930.

[24] G. R. Price, "Fisher's "fundamental theorem" made clear," *Annals of Human Genetics,* vol. 36, pp. 129-140, 1972.

[25] A. W. F. Edwards, "The fundamental theorem of natural selection". *Biological Reviews,* vol. 69, pp. 443–474, 1994.

[26] J. Neumann, G. Lohmann, J. Derrfuss, and D.Y. von Cramon, "The meta-analysis of functional imaging data using replicator dynamics," *Human Brain Mapping*, vol. 25, no. 1, pp. 165-173, 2005.

[27] P. Hammerstein, "Why is reciprocity so rare in social animals? A protestant appeal" In: P. Hammerstein, Editor, *Genetic and Cultural Evolution of Cooperation*, MIT Press. pp. 83–94, 2003.

[28] M. J. Guyer and A. Rapoport, "Information effects in two mixed motive games," *Behavioral Science*, vol. 14, pp. 467-482, 1969.

[29] D. Fudenberg and D. K. Levine, "Consistency and cautious fictitious play," *Journal of Economic Dynamics and Control*, vol. 19, no. 5-7, pp. 1065-1089, 1995.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 173 – 180

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Image Similarity Detection in Large Visual Data Bases

Juliusz L. Kulikowski
Institute of Biocybernetics and
Biomedical Engineering, Polish
Academy of Sciences
4, Ks. Trojdena Str.
02-109 Warsaw, Poland
E-mail: jkulikowski@ibib.waw.pl

***Abstract*—A method of similarity clusters detection in large visual databases is described in this work. Similarity clusters have been defined on the basis of a general concept of similarity measure. The method is based also on the properties of morphological spectra as a tool for image presentation. In the proposed method similarity of selected spectral components in selected basic windows are used to similarity of images evaluation. Similarity clusters are detected in an iterative process in which non-perspective subsets of images are step-by-step removed from considerations. In the method similarity graphs and hyper-graphs also play an auxiliary role. The method is illustrated by an example of a collection of medical images in which similarity clusters have been detected.**

## I. Introduction

VISUAL data bases (*VDB*) are widely used in various experimental investigation areas [2, 3, 5, 6, 21]. Documents stored in *VDB*s consist of a digital representation of an image (e.g. given in the form of a bit-map), of a sequence of formal data identifying the document and, possibly, of a series of qualitative and/or quantitative attributes characterizing image content. One of the main problems in *VDB* exploration is retrieval of visual documents satisfying the formal and content requirements given by the users. A typical query coming from a *VDB* user concerns all available visual documents satisfying formal (type, source, emission data, etc.) as well as content requirements [5, 20]. However, in certain cases another type of queries concerning visual documents is possible: among a class of visual documents satisfying some general formal requirements, find all subsets of documents forming, in a below-defined sense, *similarity clusters*. In this case it is assumed that a concept of *similarity* has been defined by the user (instead of defining the image content attributes). Moreover, the number of similarity clusters is beforehand neither defined nor limited, single documents as similarity clusters being out of interest. On the other hand, some documents, as it will be shown below, can be included into more than one similarity cluster. We call the above-formulated problem a *similarity clusters detection* (*SCD*) *problem*. It should be emphasized that there is a substantial difference between the well known problem of strong classification of objects [1,16] and this of *SCD*. The difference is caused, in general, by non-transitivity of similarity relation, as it can be illustrated by an example of a dactyloscopic database. The classification problem consists in this case in assigning fingerprints to similarity classes strongly defined on the basis of dactyloscopic patterns and minutes (bridges, meshes, forks, line ends, etc.). Similarity classes are pair-wise disjoint and each object belongs to exactly one similarity class. On the other hand, a *SCD* problem may consists in finding all subsets of fingerprints containing, at least, one subset of minutes of the corresponding types forming the same geometrical configuration. A visual document, due to several types of minutes satisfying the above-formulated similarity criterion, can be included into more than one similarity clusters. Solution of the *SCD* problem, as leading to a class of *NP*-complete numerical tasks, in the case of large visual databases leads to high calculation costs.

The aim of this paper is presentation of a method of reducing the calculation costs of *SCD* due to a multi-step similarity detection strategy. The strategy is based on a concept of a step-wise strengthening of similarity criteria connected with elimination of not satisfying them pairs of documents. This concept is realized due to an additional concept of using *morphological spectra* to image description; the concept was formerly used to partial similarity of documents detection [11].

The paper is organized as follows: in Sec. II basic notions of *similarity measure ε-similarity clusters* (Sec. II *A*), and of *representation of images* (Sec. II *B*) are shortly reminded. In Sec. III the concept of multi-step similarity detection is presented generally (Sec. III *A*) and an algorithm of ε-similarity clusters detection in (Sec. III *B*) is described. An example illustrating using the proposed method to a collection of medical images is presented in Sec. IV. Conclusions are formulated in Sec. V.

## II. Basic Notions

The notion of *similarity* plays a basic role in pattern recognition. In the strongest sense it can be considered as a synonym of *equivalence*, i.e. a binary relation satisfying the reciprocity, symmetry and transitivity conditions [9]. However, such similarity concept does not suit well to a description of similarity of images where the transitivity condition

978-83-60810-14-9/08/$25.00 © 2008 IEEE

is often not satisfied. In a wider sense, similarity is a sort of *neighborhood* relation (reciprocal and symmetrical) rather than this of equivalence.

*I. Similarity measures and similarity clusters*

A numerical characterization of similarity in wider sense is possible due to a *similarity measure c*oncept. This concept can be defined in several ways [9,14]. In pattern recognition similarity measure is usually defined on the basis of a *distance measure* or of a *cosine* (angular) measure [11]. That is why the following definition of similarity measure below is given (see also [9,11]):

*Definition 1.* Let *C* be a set of elements and let *a, b, c* ∈ *C* be any of its members; then a function:

$$\sigma\colon\ C\times C \to [0,\ldots,1] \qquad \textbf{(1)}$$

satisfying the conditions:

I. $\sigma(a,a) \equiv 1$,

II. $\sigma(a,b) \equiv \sigma(b,a)$,

III. $\sigma(a,b)\cdot\sigma(b,c) \le \sigma(a,c)$

will be called a *similarity measure* described on *C* ●

In particular, if *C* is a metric space [8] and *d*(*a,b*) is a distance measure of the given pair of its elements then a function:

$$\sigma(a,b) = exp\,[\,-\alpha\cdot d(a,b)], \qquad \textbf{(2)}$$

$\alpha$ being a positive scaling coefficient, satisfies the conditions of Definition 1.

Another possibility arises if *C* is assumed to be a linear unitary space [19]. In such case ***a**, **b***, etc. are interpreted as *vectors*, (***a**,**b***) denotes their *scalar product*, $\|\boldsymbol{a}\| = (\boldsymbol{a},\boldsymbol{a})^{1/2}$ is a *norm* of ***a*** and the cosine of the angle ∠(***a**,**b***) between the vectors is given by the well-known formula:

$$\cos(a,b)=\frac{(a,b)}{\|a\|.\|b\|} \qquad (3)$$

However, *cos(**a**,**b**)* cannot be used as a similarity measure satisfying the condition III of Definition 1. For this purpose it will be used the following *angular similarity measure*:

$$\sigma(a,b)=1-\sqrt{1\text{-}\cos^2(a,b)} \qquad (4)$$

On the basis of similarity measure a concept analogous to this of *similarity classes* used in equivalence (i.e. strong similarity) relation can be introduced.

*Definition 2.* Let *C* be a set of elements, $\sigma(\,a,b)$ be a similarity measure defined on *C* and let ε such that $0 \le \varepsilon \le 1$ be an arbitrary constant. Then a subset $\Phi_\varepsilon \subseteq C$ such that:

1 st for any pair of its elements $a\,,b \in \Phi_\varepsilon$ it is $\sigma(\,a,b) \ge \varepsilon$, and

2 nd for each element *c* belonging to $C \setminus \Phi_\varepsilon$ there is at least one element *a* in $\Phi_\varepsilon$ such that the inequality $\sigma(\,a,c) < \varepsilon$ is satisfied,

will be called an *ε -similarity cluster* ●

Let us remark that, excepting the case of ε = 1, there is a substantial difference between the strong *similarity class* concept and this of the *ε -similarity cluster* . It is illustrated in Fig. 1 where a set *C* consisting of 4 elements on an Euclidean plane is shown. The elements are located in the vertices of a square of unit edge-lengths. A similarity of vertices has been defined on the basis of their Euclidean distance and the *ε* - similarity clusters consist of subsets of vertices whose distance is not greater than 1. In such case it is clear that two *ε* - similarity clusters can be established in two alternative ways; such situation in strong similarity classes could not arise.

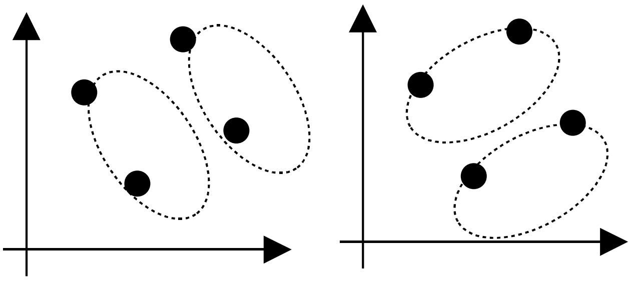


Fig. 1 . Two alternative ways of choosing ε -similarity clusters.

In similarity of objects evaluation a multi-aspect similarity can be taken into consideration by using several similarity measures. For this purpose, if $\sigma_1(a,b),\ \sigma_2(a,b),\ \ldots,\sigma_f(a,b)$ are similarity measures satisfying the conditions of Definition 1 and representing different similarity aspects then it can be shown [10] that

$$\sigma(a,b)=\prod_{\phi=1}^{f}\sigma_{\phi}(a,b) \qquad \textbf{(5)}$$

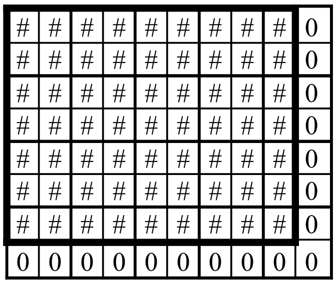
also satisfies the conditions of Definition 1 and as such it can be used as a mutli-aspect similarity measure of the given objects. This property will be used in *ε - SCD* of images.

*II. Spectral representation of images*

It will be considered representation of monochromatic images in visual databases in a basic form of *bitmaps*, i.e. numerical *I* × *J* rectangular matrices, where *I* and *J* denote, respectively, the number of rows and columns. A bitmap representing an image ***u*** will be denoted by ***U*** while its elements $u_{ij}$, called pixel values, will be assumed to be integers from a finite interval $[0,\ldots,2^k-1]$, *k* being a fixed natural number. Below, expansion of a bitmap ***U*** into a linear *I·J*-component column vector ***V***, as more convenient for calculations, will also be used.

Images can also be represented in several alternative, spectral forms [10,17]. Below, image representation by systems of morphological spectral components will be considered [12,13].

*Morphological spectra* are basically defined for monochromatic images given in the form of $2^m \times 2^m$ - size bitmaps, where *m* is a fixed natural number such that $2^m \le$ *min* ( *I,J* ). Morphological spectra form a hierarchical structure, *m* being the highest level of the hierarchy and the 0 th level being identified with the original image. The *h* -th level spectral components, where $0 \le h \le m$, are calculated on *basic windows* of $2^h \times 2^h$ size. Therefore, it is assumed that for calculation of any *h*- th level morphological spectrum the original bitmap is partitioned into a number of adjacent basic windows covering the image area.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| # | # | # | # | # | # | # | # | # | 0 |
| # | # | # | # | # | # | # | # | # | 0 |
| # | # | # | # | # | # | # | # | # | 0 |
| # | # | # | # | # | # | # | # | # | 0 |
| # | # | # | # | # | # | # | # | # | 0 |
| # | # | # | # | # | # | # | # | # | 0 |
| # | # | # | # | # | # | # | # | # | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fig. 2. Partition of an image area into basic windows.

If necessary, the image area can be covered with certain margins as shown in Fig. 2, where an image of a 7×9 (*I*= 7, *J*= 9) size has been covered by basic windows of $2^1 \times 2^1$ (i.e. *h* =1) size, the lacking pixel values in the extreme right column and in the lowest row being filled with 0-s.

In each basic window the values of a fixed *h* -th level spectral component are calculated independently. Therefore, if *p, q* denote, respectively, the number of rows and columns of basic windows covering the image ( *p* =4, *q* =5 in the above-shown case) then the values of each *h* -th level spectral component of a total image can be collected in a $p \times q$ real matrix called a *spectral component matrix* . Any *h* -th level morphological spectrum consists of $2^{2h}$ types of components. For *h* >0, spectral components are labeled by *h*- element strings consisting of symbols *Σ*, *V, H, X* ; the 1 st level morphological spectrum contains four components labeled by single symbols *Σ*, *V, H* and *X* only. The 2 nd level morphological spectrum contains $2^4$=16 components labeled and lexicographically ordered as follows: ***ΣΣ***, *Σ V, Σ H, Σ X, V Σ, VV, VH, VX, H Σ, HV, HH, HX, X Σ, XV, XH* and *XX.* The 3 rd level spectral components are denoted by ***ΣΣΣ***, ***ΣΣ*** *V*, ***ΣΣ*** *H*, … etc . The components of morphological spectra can be thus represented by a regular tree whose nodes on a given level are assigned to the given spectral-level components [17]. The spectral component labels are used to a denotation of spectral component matrices. For example, $\boldsymbol{M}_V$ , $\boldsymbol{M}_{VX}$ , $\boldsymbol{M}_{\Sigma\Sigma H}$ , etc. denote, respectively, the spectral component matrices of the spectra *V, VX* and *ΣΣ H.* For a given image size the size of the corresponding spectral component matrices depends on the spectrum level *h* and is decreasing with it. Hence, the number of elements representing a given image on each spectrum level corresponds to the number of pixels in the original image (it is exactly equal to this number if the image area can be covered without margins by the highest-level basic windows).

Morphological spectra can be calculated by using *spectral matrices* [13] . For this purpose for each (*h*-th) spectrum level it is constructed a matrix $\boldsymbol{M}^{(h)}$ of $4^h \times 4^h$ size whose rows correspond to lexicographically ordered spectral components and columns are assigned to the lexicographically ordered pixels in the basic window. For example, the 1st level morphological spectrum can be represented by a matrix:

$$M^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \qquad (6)$$

Each row of the spectral matrix consisting of the elements +1 and –1 only represents the weights assigned to pixel values whose sum within a basic window should be calculated. Spectral matrices of any level satisfy the following orthogonality condition:

$$\boldsymbol{M}^{(h)} \cdot (\boldsymbol{M}^{(h)})^{tr} = (\boldsymbol{M}^{(h)})^{tr} \cdot \boldsymbol{M}^{(h)} = 4^h \cdot \boldsymbol{I} \qquad \textbf{(7)}$$

where tr denotes matrix transposition and ***I*** is an unity matrix of $2^{2h} \times 2^{2h}$ size. If pixel values of a basic window are presented in the form of a vector:

$$\boldsymbol{V}^{(h)} = [\xi_{1,1}, \xi_{1,2}, \ldots, \xi_{1,K}, \xi_{2,1}, \ldots, \xi_{2,K}, \ldots, \xi_{K,1}, \ldots, \xi_{K,K}] \qquad \textbf{(8)}$$

where $K = 4^h$ , then the morphological spectrum of the basic window can be calculated as:

$$(\boldsymbol{W}^{(h)})^{tr} = \boldsymbol{M}^{(h)} \cdot (\boldsymbol{V}^{(h)})^{tr}. \qquad \textbf{(9)}$$

A formula for reverse bitmap components calculation from the morphological spectral components then takes the form:

$$\boldsymbol{V}^{(h)} = 4^{-h} \cdot \boldsymbol{W}^{(h)} \cdot \boldsymbol{M}^{(h)}. \qquad \textbf{(10)}$$

An important property of morphological spectra as tools for image representation consists in their multi-scalar structure and possibility to focus image description in given regions of interest covered by selected elements of the spectral component matrices. These properties will be used below in a concept of step-wise *ε-SCD* of images.

## III. Multi-step similarity clusters detection

It will be assumed that there is given a finite set *C* of images available in the form of bitmaps and a similarity measure *σ* defined on the Cartesian product *C*×*C*. The problem consists in finding in *C*, for a certain *ε*, $0 \le \varepsilon \le 1$, (usually kept close to 1) all *ε*-clusters. If *N* = |*C*| denotes the number of elements in *C* then the simplest (but very time-consuming) solution method can be based on a direct review of all ½*N*(*N*-1) pairs of analyzed objects (images). The algorithm will then consist of the following (roughly defined) steps:

*Algorithm 1 (naïve)*:

Step 1: take into consideration all unordered pairs of objects $\{ a ,b \} \in C \times C$ ;

Step 2: for each pair calculate its similarity measure $\sigma( a,b )$;

Step 3: accept all pairs for which the condition $\varepsilon \le \sigma( a,b ) \le 1$ is satisfied and reject the other ones;

Step 4: for the accepted pairs construct a set $C^*$ , $C^* \subseteq C$ , of all occurring in them objects;

Step 5: construct a *similarity graph* *G* = [ *C** , *S, φ* ] where *A* is a set of its *nodes, S* is a set of undirected *edges* , and *φ* is a function assigning, in an unique way, to each accepted pair of nodes an edge from the set *S* ;

Step 6: using a standard algorithm of finding *cliques* in graphs find all cliques in *G* ;

Step 7: end ●

The cliques (maximal complete subgraphs) of the similarity graph *G* found by the algorithm are, by the same, the ε-similarity clusters. On the other hand, it should be remarked that if *F* is a family of cliques of the similarity graph *G* then a triple:

$$H = [C^*, F, \psi] \qquad \textbf{(11)}$$

defines a hyper-graph in the Berge sense [7], where $C^*$ is a set of nodes, ***F*** is a set of hyper-edges and $\psi$ is a partial function described on the family $2^{C^*}$ of the subsets of $C^*$, assigning in an unique way the hyper-edges of *F* to selected subsets of nodes. We shall call *H* a *similarity clusters hyper-graph*; its role in *SCD* algorithms will be shown below.

However, reducing a *SCD* problem to this of a classical *cliques* finding problem is practically not a satisfying task solution because of its non-polynomial (*NP*) complexity. Despite the fact that there are several approximate algorithms for it described in the literature [4,15], an approach to the calculation cost reduction based on a step-wise restricting of ε-similar cliques is proposed below.

*I. General concept description*

For the purpose of comparison of images they will be considered as ordered sets of sub-bitmaps according to their partition into basic windows, as shown in Fig. 2. A bitmap ***U*** of a given image ***u*** will be thus represented by a composed matrix of sub-bitmaps:

$$U = [U_{\kappa\lambda}] \qquad (12)$$

where $U_{\kappa\lambda}$ is a sub-bitmap, $1\le\kappa\le p$, $1\le\lambda\le q$ . Similarity of any given pair ***u', u"*** of images means that the pairs of the corresponding sub-bitmaps $U'_{\kappa\lambda}$ , $U''_{\kappa\lambda}$ satisfy some similarity criteria. Similarity of any given pair of corresponding sub-bitmaps can be thus considered as an aspect of similarity of the images ***u', u"*** in the whole. According to (5), their similarity measure can be expressed as

$$\sigma(U',U'')=\prod_{\kappa=1}^{p}\prod_{\lambda=1}^{q}\sigma(U'_{\kappa\lambda},U''_{\kappa\lambda}) \qquad (13)$$

Let us denote by *L* , $L = \{(\kappa,\lambda)\}$, the set of all considered pairs $(\kappa,\lambda)$. It is clear that due to the general properties of similarity measures the inequalities

$$0\le\sigma(U',U'')\le min_L[\sigma(U'_{\kappa\lambda},U''_{\kappa\lambda})] \qquad \textbf{(14)}$$

are held. Therefore, if for a given pair of images ***u', u"*** it is required that $\sigma(U',U'')\ge\varepsilon$ and for a certain pair $(\kappa,\lambda)\in L$ it has been found that $\sigma(U'_{\kappa\lambda},U''_{\kappa\lambda})<\varepsilon$ then there is no reason to prove the similarity condition in other pairs of sub-bitmaps and the given pair ( ***U', U"*** ) can be considered as "non-perspective" from its ε -similarity point of view. Moreover, if we denote by *L'* , $L'\subseteq L$ a certain subset of pairs $(\kappa,\lambda)$ of indices for which it has been found that

$$\sigma(U',U'')=\prod_{(L')}\sigma(U'_{\kappa\lambda},U''_{\kappa\lambda})=\varepsilon' \qquad \textbf{(15)}$$

where $\varepsilon'$ is a number $\ge\varepsilon$ then for satisfying the condition $\sigma(U',U'')\ge\varepsilon$ it is necessary that an inequality

$$\sigma_{L\setminus L'}(U',U'')=\prod_{(L\setminus L')}\sigma(U'_{\kappa\lambda},U''_{\kappa\lambda})\ge\frac{\varepsilon'}{\varepsilon} \qquad \textbf{(16)}$$

is satisfied.

For similar reasons, for a fixed pair of sub-bitmaps ($U'_{\kappa\lambda}$, $U''_{\kappa\lambda}$) their similarity measure can be considered as a multi-aspect similarity measure of the corresponding pairs of morphological spectra

The below-proposed procedure of ε-*SCD* of images is based on a concept of step-by-step strengthening of similarity criteria connected with removing non-perspective pairs of images from considerations and improving accuracy of the *SCD*.

*II. Choosing the ε-similarity thresholds*

At an initial state of the procedure it is posed the following

*Initial working hypothesis*: the similarity measures of the pairs of all corresponding basic windows and spectral components of all pairs (***u', u"***) of images in the analyzed set *C* are equal 1.

By this assumption, the condition $\sigma(u',u'')\ge\varepsilon$ according to (5) and (14) is satisfied and *C* constitutes, as a whole, the first assumed, hypothetical ε -similarity cluster.

In the consecutive iterations of the procedure the working hypothesis by evaluation of similarity measure of selected pairs of basic windows and spectral components (selected similarity aspects) is verified. The condition $\sigma(u',u'')\ge\varepsilon$ is satisfied if it is satisfied by all pairs of corresponding basic windows and spectral components. As a consequence:

i. the objects whose similarity to all other objects does not satisfy the condition $\sigma(u',u'')\ge\varepsilon$ can be removed from considerations as non-perspective ones;

ii. the pairs of perspective (non-removed) objects whose similarity does not satisfy the above-given condition are taken into consideration; however, they bring about a necessity of correction of the currently assumed, hypothetical similarity clusters;

iii. at each iteration of the procedure the similarity threshold levels should be corrected according to the general principle described in Sec. *A* ;

iv. at each iteration the similarity measure of basic windows and of spectral components which yet have not been evaluated remain equal 1;

v. each iteration (excepting the last one) leads to a revised subset of hypothetical similarity clusters which in the next iteration in similar way should be processed.

### *III. The procedure of $\varepsilon$-similarity clusters detection*

For a given initial set $C$ of images and a fixed final similarity measure value $\varepsilon$, $0 < \varepsilon \leq 1$, a morphological spectrum level $h$ determining the size of basic windows should be chosen by taking into account that the larger is this size the higher is the probability that differences between the images, if any exist, by the algorithm in each iteration will be detected. On the other hand, the higher is $h$ the larger is the number of spectral components that should be taken into account in the *SCD* algorithm.

The $\varepsilon$-*SCD* procedure consists of a sequence of iterations, each iteration consisting of two phases:

1st reduction, according to the strengthened similarity criteria, of the set of compared pairs of objects;

2nd decomposition and/or reduction of the similarity clusters inherited from the former iteration.

Each ($i$-th) iteration of the algorithm needs the following data to be entered:

i. a starting threshold level $\varepsilon^{(i)}$, $0 < \varepsilon < \varepsilon^{(i)}$;

ii. a pair$\{(\kappa^{(i)}, \lambda^{(i)})\}$ of addresses of basic windows selected for being used in the current iteration of *SCD* procedure;

iii. a label $\boldsymbol{\Gamma}^{(i)}$ (subset of labels) of the $h$- th level morphological spectrum component selected for being used in the current iteration of *SCD* procedure.

No strong rules of choosing the above-mentioned data exist. However, the following, heuristic recommendations can be taken into account:

a) according to the formula (16), for any $i$ =1,2,… the threshold level $\varepsilon^{(i)}$ should be chosen as $\varepsilon^{(i)} = \varepsilon^{*(i-1)}/\varepsilon$ where $\varepsilon^{*(i-1)}$ denotes the similarity measure of the given pair of images evaluated in the former, $(i - 1)$ iteration;

b) the basic windows selected for analysis should be, if possible, selected with a preference given to the components having the highest discriminative power.

To meet the recommendations the following, auxiliary objects are defined:

* a symmetric square matrix $\boldsymbol{S} = [\sigma^{(i)}_{\alpha\beta}]$ of $N\times N$ size, $N$ denoting the number of elements of $C$; elements $\sigma^{(i)}_{\alpha\beta}$ denote the similarity measures of the pairs of bitmaps $(\boldsymbol{U}^{(\alpha)},\boldsymbol{U}^{(\beta)})$ evaluated at the $i$-th iteration);

* a binary matrix $\boldsymbol{M} = [m^{(i)}_{\beta\gamma}]$, $\beta$ denoting a serial number of basic window (equivalent to its $(\kappa,\lambda)$ address), $\gamma$ being assigned to a serial number of spectral component; $m^{(i)}_{\beta\gamma} = 1$ if the $\beta$-th basic window and the $\gamma$- th spectral component have been already used to the similarity of images assessment, otherwise $m^{(i)}_{\beta\gamma} = 0$.

At the initial state of the procedure all elements of $\boldsymbol{S} = \boldsymbol{S}^{(0)}$ equal 1 and all elements of $\boldsymbol{M} = \boldsymbol{M}^{(0)}$ equal 0.

In the below-presented concept of algorithm a graph representation of the current state of the $\varepsilon$-*SCD* procedure is also useful.

A similarity graph $G$ (see Sec. III) is described by:

a) a subset $C^* \subseteq C$ of nodes representing the images currently considered as "perspective" for $\varepsilon$ - *SCD* ;

b) a symmetrical square sub-matrix $\boldsymbol{S^*} \subseteq \boldsymbol{S}$ containing the rows and columns of $\boldsymbol{S}$ corresponding to "perspective" images, the elements of $\boldsymbol{S^*}$ being interpreted as weighed edges of the graph $G$ . $\boldsymbol{S^*}$ plays the role of an adjacency matrix of $G$ .

At a starting point the similarity graph $G = G^{(0)}$ is assumed to be a complete graph identical to its unique clique.

A current state of the $\varepsilon$ - *SCD* task solution can also be illustrated by an $\varepsilon$ - *similarity cluster hyper-graph $H$* . Its definition is given by the formula (11) excepting that $F$ denotes a family of hyper-edges representing the currently detected, hypothetical $\varepsilon$ - *similarity* cliques $\Phi_s$ , $s$=0,1,2,3,…, in $G$. At a starting point the hyper-graph $H = H^{(0)}$ contains a single hyper-edge: $F^{(0)} = \{\Phi_0\}$, $\Phi_0 \equiv C^*$.

For similarity of pairs of images assessment the similarity measure of spectral vectors based on the formulae (2) or (4) and (5) can be used. A progress index $z$ also will be used to mark the situations when in the $i$-th iteration the necessity of similarity clusters hyper-graph $H$ correction has been detected.

The $i$-th iteration, $i$ = 1,2,3,…, of the *SCD* algorithm applied to a hypothetical $\varepsilon$-similarity cluster detected in a previous iteration has the following structure:

*Algorithm 2 (similarity clusters specifying)*:

```
Step 1: set the initial data values: $\varepsilon^{(i)}$, $z := 0$, $\boldsymbol{S}^{(i)} := \boldsymbol{S}^{(i-1)}$ , $\boldsymbol{M}^{(i)} := \boldsymbol{M}^{(i-1)}$ and $\boldsymbol{F}^{(i)} := \boldsymbol{F}^{(i-1)}$;
Step 2: select the next basic window $(\kappa,\lambda) := (\kappa^{(i)}, \lambda^{(i)})$ and the next spectral component $\boldsymbol{\Gamma} := \boldsymbol{\Gamma}^{(i)}$ from the set of non-zero elements of $\boldsymbol{M}^{(i)}$;
Step 3: according to the selection made in Step 2 set $m^{(i)}_{\beta\gamma} := 1$;
Step 4: for all non-zero elements $\sigma^{(i-1)}_{\alpha\beta}$ of adjacency matrix $\boldsymbol{S}^{(i)}$:
    1) select the sub-bitmaps $\boldsymbol{U}^{(\alpha)}_{\kappa\lambda}$ , $\boldsymbol{U}^{(\beta)}_{\kappa\lambda}$ corresponding to the basic windows $(\kappa^{(i)},\lambda^{(i)})$;
    2) for $\boldsymbol{U}^{(\alpha)}_{\kappa\lambda}$ , $\boldsymbol{U}^{(\beta)}_{\kappa\lambda}$ and the given $\boldsymbol{\Gamma}$ calculate the spectral components $\boldsymbol{W}^{(\alpha)}_{\Gamma;\kappa\lambda}(\alpha)$, $\boldsymbol{W}^{(\beta)}_{\Gamma;\kappa\lambda}$;
    3) calculate the similarity measure values $\sigma^{(i)}_{\alpha\beta} = \sigma[\boldsymbol{W}^{(\alpha)}_{\Gamma;\kappa\lambda}, \boldsymbol{W}^{(\beta)}_{\Gamma;\kappa\lambda}]$ using the formula (2);
Step 4: if $\sigma^{(i)}_{\alpha\beta} = \sigma[\boldsymbol{W}^{(\alpha)}_{\Gamma;\kappa\lambda}, \boldsymbol{W}^{(\beta)}_{\Gamma;\kappa\lambda}] \geq \varepsilon^{(i)}$ then in $\boldsymbol{S}^{(i)}$ set $\sigma^{(i)}_{\alpha\beta} := \sigma[\boldsymbol{W}^{(\alpha)}_{\Gamma;\kappa\lambda}, \boldsymbol{W}^{(\beta)}_{\Gamma;\kappa\lambda}]$, otherwise set $\sigma^{(i)}_{\alpha\beta} := 0$ and $z := 1$;
Step 5: if $z = 0$ then go to Step 8, otherwise:
Step 6: if all elements $\sigma^{(i)}_{\alpha\beta}$ of the $\alpha$ - row (respectively, of the $\beta$-row) of $\boldsymbol{S}^{(i)}$ equal 0 then remove the corresponding row and column and remove $\boldsymbol{u}^{(\alpha)}$ (respectively, $\boldsymbol{u}^{(\beta)}$) from $C^*$, and from all containing it hyper-edges in $\boldsymbol{F}^{(i)}$ , then go to Step 8, otherwise:
Step 7: in the set $\boldsymbol{F}^{(i)}$ of hyper-edges:
    1) find all hyper-edges $\Phi^{(i)}_m$ containing both nodes $\boldsymbol{u}^{(\alpha)}$ and $\boldsymbol{u}^{(\beta)}$;
    2) replace each hyper-edge $\Phi^{(i)}_m$ found in 1)
       by two hyper-edges: $\Phi^{(i)'}_m := \Phi^{(i))}_m \setminus \{\boldsymbol{u}^{(\alpha)}\}$ and
       $\Phi^{(i)''}_m := \Phi^{(i))}_m \setminus \{\boldsymbol{u}^{(\beta)}\}$;
    3) remove $\Phi^{(i)'}_m$ (respectively, $\Phi^{(i)''}_m$) from $\boldsymbol{F}^{(i)}$ if
       it consists of one node only;
    4) assign to the hyper-edges new indexes unifying the enumeration in the set $\boldsymbol{F}^{(i)}$ ;
  Step 8: check whether all elements of $\boldsymbol{M}^{(\mathrm{i})}$
equal 1; if no then go to Step 1, otherwise
  Step 9: end of the algorithm ●
```

Algorithm 2 should be applied to all $\varepsilon$-similarity clusters found in the previous iterations of the ε-*SCD* procedure.

## IV. Application to Medical *VDB* Exploration

Large medical *VDB* s may contain thousands of visual documents of various modalities stored in hospitals, specialized medical clinics or in departments and laboratories of medical universities. The *VDB* exploration problems usually concern retrieval of documents based on their formal features or contents. However, in scientific investigations as well as in difficult diagnostic problems *SCD* can also play a significant role.

### *I. Example*

In Fig. 3 a sample of a large collection of typical human brain images obtained by SPECT (Single Photon Emission Tomography) technique is shown. The images are of 124 × 124 pixels size partitioned into basic windows of 16 × 16 pixels size arranged in columns (denoted by numbers from *1* to *8*) and in rows (denoted by symbols from *A* to *H*). The contents of a single basic window can be represented by an 8 × 8 size bitmap or by its 3rd level morphological spectrum consisting of 64 components.

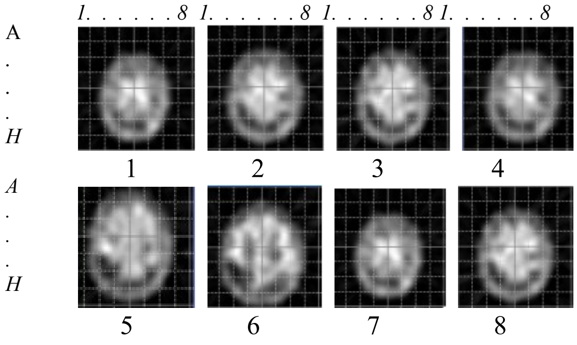


Fig. 3. Cerebral SPECT images stored in a *VDB* .

Fig. 4 shows intensity maps of several spectral components of an image are (symbol *S* stands here for $\Sigma$ ). For comparison of images a region of interest (*ROI*) consisting of *K*, *K* ≤ 64, basic windows not belonging to the background should be considered.

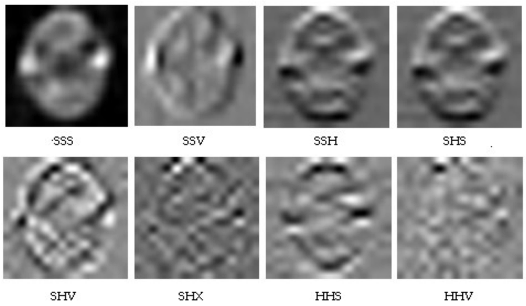


Fig. 4. Selected examples of spectral components' intensities of a given image.

Looking at Fig. 3 it can be remarked that, probably, the subsets of images: {#1, #7} and {#2, #4, #8} form similarity clusters, while images #5 and #6 do not match to any other ones. However, this observation is not based on an objective similarity measure, hence, for a more precise task solution a 0.9-*SCD* problem will be considered.

First, *ROI*s identical in size and form in the set of images should be chosen. Next, a similarity measure based on a distance measure of vectors:

$$d(v', v'') = \sum_{(k)} |v'_k - v''_k| \qquad (17)$$

as the simplest one will be chosen.

In the case of using Algorithm 1 each of *K* (*K* ≤ 64) basic windows containing 64 pixels, should be compared with respective basic windows of all other images. Then, all pairs of vectors not satisfying the inequality $\sigma(\boldsymbol{v'},\boldsymbol{v''}) \geq 0.9$ should be removed, a similarity graph *G* consisting of 8 nodes and of all edges satisfying the given inequality should be constructed and, according to, the cliques finding problem should be solved.

An alternative, on Algorithm 2 based approach to the problem consists in its iterative solution connected with step-wise elimination of non-perspective pairs of vectors.

At the first iteration the basic windows (*D*,*4*) in the images using the $\Sigma\Sigma\Sigma$ (a sum of 64 pixel values filling a single basic window) spectral component will be compared. As a result, it can be established that the similarity between image #6 and all other ones is below the threshold 0.9 and #6 should be removed from considerations. The problem is thus reduced to this of finding cliques in a complete similarity graph $G^{(1)}$ consisting of 7 nodes: #1, #2, #3, #4, #5, #7 and #8. $G^{(1)}$ is by assumption identical to its unique clique; however, this working hypothesis by testing the similarity measure of selected pairs of basic windows and spectral components should verified.

At the second iteration of the procedure for testing there have been selected (in principle, randomly) the (*F,6*) basic windows and the $\Sigma HV$ spectral component. As a result two further dissimilarities between the pairs of vectors corresponding to the images (#1,#2) and (#1,#4) have been detected. Therefore, value 0 should be assigned to the elements $\sigma_{12}$, $\sigma_{21}$, $\sigma_{14}$ and $\sigma_{21}$ of the adjacency matrix $\boldsymbol{S}^{(2)}$ of the similarity graph $G^{(2)}$. Moreover, the non-perspective edges

(#1,#2) and (#1,#4) of the graph lead to its replacement by maximal complete sub-graphs non-containing the prohibited pairs of nodes. This means that $\boldsymbol{S}^{(2)}$ should be replaced by two its sub-matrices $\boldsymbol{S}^{(2,1)}$ and $\boldsymbol{S}^{(2,2)}$ based, respectively, on the following rows (columns): {#2, #3, #4, #5, #7, #8} and {#1, #3, #5, #7, #8}. Consequently, the similarity hypergraph $H^{(2)}$ will consist of the set of 7 nodes:

$$C^* = \{\#1, \#2, \#3, \#4, \#5, \#7, \#8\}$$

and of the set of hyper-edges:

$$F = \{(\#2, \#3, \#4, \#5, \#7, \#8),\ (\#1, \#3, \#5, \#7, \#8)\}.$$

Next iterations will be based on increased threshold levels $\varepsilon^{(i)}$ ; the results will be presented after removing single-node subsets and subsets included by some larger ones. Let us select for being tested the basic windows ( $F$ ,3) and spectral components $\Sigma H \Sigma$. Then dissimilarity of the pairs (#2, #7), (#4, #7), (#5, #7) and (#7, #8) can be detected. This preserves the former set $C^*$ of nodes but it leads to the following hypothetical set of hyper-edges:

$$F = \{(\#2, \#3, \#4, \#5, \#8), (\#1, \#3, \#7), (\#1, \#3, \#5, \#8)\}.$$

Next iteration, based on the components ($D$,6), $\Sigma HX$, detects dissimilarity of the pairs (#1, #3), (#1, #4), (#1, #5), (#1, #8), (#2, #5), (#2, #8), (#3, #5) and (#4, #5). This leads to the set of hyper-edges:

$$F = \{(\#3, \#4, \#8), (\#2, \#3, \#4), (\#1, \#7), (\#3, \#7), (\#3, \#5, \#8)\}.$$

### *II. Comments*

In the above-presented example the results have been reached due to comparison of several basic windows and spectral components only instead of comparing full images.

The procedure is stopped when in an iteration all $\varepsilon$ -similarity clusters disappear or if the number and form of the clusters does not change for several iterations.

## V. Conclusions

The following properties of the above-described method of $\varepsilon$ - $SCD$ should be remarked:

i. Each iteration of the algorithm consists of a finite number of repetitions of a finite sequence of steps.

ii. The number of iterations is finite.

iii. Each iteration leads to an approximation of the solution satisfying more rigid ε-similarity criteria, hence it preserves or reduces the size of formerly found ε -similarity clusters.

iv. Each next iteration consists in finding cliques in a similarity graph $G$ containing a non-increased (in most cases – reduced) set of nodes and set of edges.

v. In the less favorable case the algorithm leads to an exact $\varepsilon$-$SCD$ task solution in a finite number of steps.

Hence, the properties of the Algorithm 2 do not guarantee that the calculation cost of an $\varepsilon$-$SCD$ task solution is in each case lower than if Algorithm 1 type one is used. However, it does guarantee that in most cases it is lower due to the reduction of the number of nodes and edges in the similarity graph $G$ leading to an exponential reduction of the cliques finding calculation cost. An exact The above-presented images are of 124×124 pixels size partitioned into basic windows of 16×16 pixels size arranged in columns (denoted by numbers from *1* to *8*) and in rows (denoted by symbols from *A* to *H*). The contents of a single basic window can be represented by an 8×8 size bitmap or by its 3rd level morphological spectrum consisting of 64 components. evaluation of the real gain in calculation cost reduction is not possible to be done by analytical methods. It needs a series of experiments to be performed on statistically representative sets of large (i.e. hundreds of nodes containing) similarity graphs.

## Acknowledgement

I owe the medical images used in Sec. IV *A* to the courtesy of Prof. MD. L. Krolicki, Head of the Department of Nuclear Medicine of the Warsaw Medical University and of Dr M. Przytulska from Dept. II of the Institute of Biocybernetics and Biomedical Engineering PAS,Warsaw. This work was partially supported by Ministry of Science and Higher Education of Poland, project nr. 4211/BT02/2007/33.

## References

[1] Aivazyan S. A., Buchstaber V. M., Yenyukov I. S., Meshalkin L. D., *Applied Statistics. Classification and Reduction of Dimensionality* (in Russian), Moscow: Finansy I Statistika, 1989.

[2] Apers P., Blanken H., Houtsma M., *Multimedia Databases in Perspective*. New York: Springer-Verlag, 1997.

[3] Arisawa H., Catarci T., Eds., *Advances in Visual Information Management. Visual Database Systems* . Kluwer Academic Publishers, Boston Dordrecht London, 2000.

[4] Auguston J. G., Minker J., "An Analysis of Some Graph-Theoretical Cluster Techniques." *J. ACM* 17(4), 1970, pp. 571-588, Errata: *J.ACM* 19(4) , 1972, pp. 244-247.

[5] Baeza-Yates R., Ribeiro-Neto B., *Modern Information Retrieval* . ACM-Press, New York, Addison-Wesley, Harlow Eng. Reading Mass., 1999.

[6] Bakker A. R., "HIS, RIS and PACS." *Comp. Med. Imag. Graph*, No 15, 1991, pp. 157-160.

[7] Berge C.. *Graphs and Hypergraphs.* Amsterdam: North-Holland, 1973.

[8] Duda R.O., Hart P. E., Stork D.G. *Pattern Classification and Scene Analysis* . New York, John Wiley & Sons, 2000.

[9] Kulikowski J. L., "Recognition of Similarities in Image Databases", *17th International CODATA Conference, Book of Abstracts,* Baveno, 2000, pp. 31-32

[10] Kulikowski J. L., "Pattern Recognition Based on Ambiguous Indications of Experts," in *Komputerowe Systemy Rozpoznawania KOSYR'2001,* Kurzyński M., Ed., Wrocław: Wyd. Politechniki Wrocławskiej,, 2001, pp. 15-22.

[11] Kulikowski J. L., Przytulska M., "Partial Similarity Based Retrieval of Images in Distributed Database", in *Advances in Intelligent Web Mastering, AWIC'2007,* Wegrzyn-Wolska K., Szczepaniak P. S., Eds. Berlin, Heidelberg, New York: Springer, 2007, pp. 186-191.

[12] Kulikowski J. L., Przytulska M., Wierzbicka D., "Recognition of Textures Based on Analysis of Multilevel Morphological Spectra", *GESTS Intern. Trans. on Computer Science and Eng,* 38(1), 2007, pp. 99-107.

[13] Kulikowski J. L., Przytulska M., Wierzbicka D., "Morphological Spectra as Tools for Texture Analysis", in *Computer*

*Recognition Systems 2* , Kurzynski M., Puchala E., Wozniak M, Zolnierek A., Eds, Berlin, Heidelberg, New York: Springer, 2007, pp. 510-517.

[14] Marek T., *Cluster Analysis in Empirical Investigations. SAHN Methods* (in Polish). PWN, Warsaw, 1989.

[15] Mulligan G. D., “Algorithm for Finding Cliques of a Graph.” *Tech. Report* No 41, Toronto: Univ. of Toronto, 1972.

[16] Noworol C., *Cluster Analysis in Empirical Investigations Fuzzy Hierarchical Models* (in Polish). Warsaw: PWN, 1989.

[17] Pratt W.K., *Digital Image Processing* . New York: John Wiley & Sons, 1978.

[18] Rasiowa H., Sikorski R., *The Mathematics of Metamathematics.* Warsaw: PWN, 1968.

[19] Reinhardt F., Soeder H., „ *dtv-Atlas Mathematik* “. Munich: Deutscher Taschenbuch Verlag, 1977.

[20] 20. Salton G., McGill M. J., *Introduction to Modern Information Retrieval.* New York: McGraw-Hill Book Co., 1983.

[21] Wong S. T. C., Ed. *Medical Image Databases.* Kluwer Academic Publishers, Boston Dordrecht London, 1998.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Automatic Acquisition of Wordnet Relations by the Morpho-Syntactic Patterns Extracted from the Corpora in Polish

Roman Kurc, Maciej Piasecki
Institute of Applied Informatics, Wrocław University of Technology, Poland
Email: maciej.piasecki@pwr.wroc.pl

***Abstract*—In the paper we present an adaptation of the Espresso algorithm of the extraction of lexical semantic relation to specific requirements of Polish. The introduced changes are of more technical character like the adaptation to the existing Polish language tools, but also we investigate the structure of the patterns that takes into account specific features of Polish as an inflectional language. A new method of the reliability measure computation is proposed. The modified version of the algorithm called Estratto was compared with the more direct reimplementation of Espresso on several corpora of Polish. We tested the influence of different algorithm parameters and different corpora on the received results.**

## I. Introduction

STARTING construction of a system from scratch gives much more control over it, but in the case of large, practical systems it usually means that it will never be completed. In current-day software engineering, the component-based architecture and re-usable components became a typical way of construction. In the contemporary Computational Linguistics and Natural Language Engineering a similar role is played by basic language resources and tools. There are attempts to define a basic set of them, e.g. [1][2], or to build architectures supporting their application, e.g. (Clarin). Wordnets[1] built for different languages became commonly applied as the source of linguistic knowledge. The main problem of the basic language resources is that they do not exist for many languages and their construction takes a lot of time and is costly. The construction of the first Polish Wordnet, called *plWordNet* (Polish name: *Słowosieć*) started in the year 2005, and its current version includes 14 677 lexical units (henceforth LUs)—one word or multiword lexemes [4]. In wordnet, LUs which are near synonyms are grouped into synsets, sets of near synonyms, and synsets are linked by lexical relations of several types. In plWordNet, the synset relations can be mapped onto the level of LU relations. One of the most important relations for any wordnet is hypernymy—simplifying, an LU $a$ is the hypernym of the $b$ if $b$ is a kind of $a$ on the basis of their lexical meanings.

The present size of plWordNet is too small for many applications. It could not be larger, since its manual construction was quite expensive. However, knowing that, from the very beginning, we assumed the manual work would be supported by the developed tools for the automatic extraction of instances of lexical semantic relations from corpora. We paid special attention to hypernymy due to its importance, i.e. our aim is to construct a tool acquiring pairs of: hypernym and hyponym from large corpora.

There are two possible paradigms [5]: *pattern-based* and *clustering based* also called *distributional*. The latter results in good recall but there are problems with precision, as its typical product is a *measure of semantic relatedness*, not some lexical semantic relationship itself. For any pair of LUs their level of relatedness can be obtained, but it is very unclear how to perform the identification of an type of relation. In the area of the clustering based paradigm, several works were done for Polish, e.g. [6], [7]. However, the number of works done in the area of pattern-based paradigm is very small, e.g. (Dernowicz, 2007), (Ceglarek & Rutkowski, 2006) the latter one dealing with the machine readable dictionaries, not corpora.

Pattern-based approaches are claimed to express good precision, but very small recall in the case of patterns constructed manually, e.g. [8]. The recall of patterns can be increased by using many or more generic patterns extracted automatically from a corpus, i.e. patterns which have broad coverage but intrinsically low precision. The system Espresso presented by Pantel and Pennacchiotti [5] is so successful an example of such an approach, that it inspired us to adapting this type of approach to Polish.

Our goal was to develop a statistical method of the extraction of lexico-morphosyntactic patterns for the needs of automatic hyperonymy acquisition. Our starting point was the adaptation of the Espresso algorithm to the Polish language, and even more important, to a very limited set of language tools for Polish. In the paper we present the measures introduced in Espresso, elements that should be taken into account when Polish patterns are being extracted and application of Espresso to extracting hypernymy. We also discuss the possibility of acquiring other types of relations. On the basis of the collected experience, an extended version of Espresso is proposed called *Estratto*.

## II. Espresso

Espresso is thought to solve the *bootstrapping* problem [9], [10] i.e. learning the structure of the domain, where

[1] A wordnet is an electronic thesaurus of the structure following the main lines of the Princeton WordNet [3]

> “the phenomena and the rules defined in terms of those categories are learned from scratch [...]”

and

> “the specification of a set of rules presupposes a set of categories, but the validity of a set of categories can only be assessed in the light of the utility of the set of rules that they support.”

So both rules and categories must be derived together. In Espresso rules are patterns and categories are semantic relations represented by *instances* defined as pairs of LUs. The algorithm consist of three phases:

- *construction* of patterns on the basis of *instances* of a relation,
- pattern statistical *evaluation*,
- and *extraction* of instances on the basis of positively evaluated patterns.

Pantel and Pennacchiotti claim [5] that Espresso is characterized by:

- high recall together with a small decrease in precision of extracted instances,
- autonomy of work (weakly supervised algorithm)—only several initial instances of the given relation must be defined at the beginning,
- independence from the size of the used corpus or a domain,
- wide range of relation types that can be extracted.

The small decrease in accuracy results from the application of generic patterns together with specific ones. This balance is achieved by the proposed measure evaluating *reliability* of patterns and instances, explained in Section *Reliability and confidence measures*. *Confidence* of instances extracted by the generic patterns is verified on a large separated corpus. The confidence of an instance originates from its strength of association with reliable patterns and the number of reliable patterns which extract it. Only the best patterns and instances are kept for the following phase of the algorithm

The introduced measure of reliability and confidence reduce the need for manual supervision once Espresso started. The measures are a means of creating rankings of instances and patterns defining the degree to which they express the target relation.

As the system of measure, instances and pattern selection are universal and do not refer to any properties of any particular relation being extracted, Espresso can be applied to a wide range of relation, and was to several, e.g. hypernymy, meronymy, antonymy but also more specific like person—company or person—job title [5].

The characteristics of Espresso inspired us to try to adapt it to Polish. We are going to investigate the key of the algorithm like measures of reliability and confidence, methods of pattern extraction and the usage of verifying corpus for the evaluation of confidence in relation to the characteristic features of Polish.

## III. Reliability and confidence measures

The reliability measure that is applied to construct the ranking of patterns and instances is one of the most important elements of the algorithm and is defined for patterns in the following way:

$$r_\pi = \frac{\sum_{i \in I} \left( \frac{pmi(i,p)}{max_{pmi}} * r_t(i) \right)}{|I|} \tag{1}$$

where $p$ is a pattern, $i$—an instance, $r_t$—measure of reliability for instances, $pmi$—Pointwise Mutual Information, explained below, and $|I|$—the size of the set of instances.

The reliability of instances is defined in a very similar way, but this time the reliability of patterns is utilized in the equation.

PMI measure originates from the Theory of Information and is defined as following:

$$pmi(i,p) = \log \frac{|x,p,y||*,*,*|}{|x,*,y||*,p,*|} \tag{2}$$

where $|x,p,y|$ is the number of occurrences of $x$ and $y$ in contexts matching the pattern $p$, $x,*,y$—the number of co-occurrences of $x$ and $y$ in the corpus regardless the pattern, etc.

The $pmi$ definition given in [5] does not include the constituent: $|*,*,*|$, i.e. the number of contexts. However, the PMI measure should be usually greater than 0, while the one defined in 2 is not. Moreover, the missing constituent is suggested also by the general definition of PMI:

$$pmi(i,p) = \log \frac{p(I,P)}{p(I)p(P)} \tag{3}$$

Because PMI is significantly greater in the situation in which instances and patterns are not numerous (e.g. the size smaller than 10), PMI is multiplied by a factor proposed in [11].

The measure of confidence of an instance extracted by the generic patterns is based on the application of specific patterns of high reliability to a different validating corpus and is calculated in the following way:

$$S(i) = \sum_{p \in P_R} S_P(i) * \frac{r_\pi(p)}{T} \tag{4}$$

$P_R$ is the set of specific patterns, $S_p = pmi(i,p)$ and $T$ is the sum over the reliability of specific patterns.

It is worth to emphasize that patterns are evaluated not on the basis of instances which were extracted by them, but on the basis of instances that were used to acquire these patterns. Instances are evaluated in a similar way. This is a consequence of the method assumed in Espresso: patterns are not matched to the instances but are induced by the instances.

The intuition behind the measures of reliability and confidence is that patterns which well describe the given relation frequently occur with a large number of confident instances of this relation. The same applies in the opposite way. However, in the case of confidence the difference is that instances extracted by generic patterns will obtain high confidence, if they occur in contexts matched by the specific patterns of good reliability in the validating corpus.

There are two unclear issues in the picture presented above. Firstly, even making a draft calculation, we can check, that reliability is sensitive to possible fluctuations in PMI value. Occurrence of higher PMI values (e.g. originating from small frequencies, even after correction by the discounting factor dependent on the number of occurrences) can cause lower assessment of patterns with balanced ratio of co-occurrence with matched instances in relation to the pattern occurrences and occurrences of instances alone. Such a situation results in the artificially increased value of $max_{pmi}$. Thus, we would like to look for a measure which would be more insensitive to the problem of the low frequency of pattern matches or instances matched. Secondly, the reliability measure of the best instance or pattern may take the value lower than one even in a situation in which there is a complete match of all patterns and all instances (or the other way round, depending on which we are calculating the reliability). That is why, the reliability propagation to the subsequent phases causes, that new values calculated for patterns on the basis of instances, and vice versa, will be gradually lower according to the size of the set for which the reliability is computed, i.e. patterns or instances. Thus we want to introduce a new measure of reliability, which returns one as the value for the best patterns or instances in every phase.

$$r_\pi(p) = \frac{\sum_{i \in I}(pmi(i,p) * r_t(i)) * d(I,p)}{max_P(\sum_{i \in I}(pmi(i,p) * r_t(i))) * |I|} \qquad (5)$$

where $d(i,p)$ defines how many unique instances the given pattern is associated with.

PMI in the formula (5) is usually modified by the discounting factor, as well.

## IV. Patterns

As for the Machine Learning methods the choice of features for objects is a very important decision, so is the choice of pattern structure and pattern language for the pattern-based approaches in acquisition of lexical semantics. The majority of approaches take the scheme proposed by Hearst [12] as their reference point, i.e. patterns being a subset of regular expressions, in which the alphabet includes lemmatized word forms and a set of variables for noun phrases matched as an element of a relation instance. An example for hipernymy could be: NP such as NP1*
NP is a/an NP1

We assumed that patterns are a subset of regular expressions with Kleene closure, but without grouping. The alphabet includes morphological base forms of lexical units. Before presenting the scheme of a pattern for Polish, we need to investigate the characteristic features of Polish, which we considered.

### A. Selected aspects of Polish

The vast majority of pattern-based approaches were developed for English. Those approaches base in some extent on the positional, linear syntactic structure of English. It was not clear how one can successfully transform this approach to a language of a significantly diffrent type like Polish.

The basic, unmarked order of a Polish sentence is Subject—Verb—Object, i.e. similarly to English. So, for simple lexico-syntactic patterns based on relative positions of described elements the differences should not be large. However, on the other side, in order to fully explore the potential of pattern-based approach we have to go beyond the analysis of the most simple constructions only. One needs to take into account such phenomena like morphosyntactic agreement of different kinds among word forms and the relaxed order of a sentence, according to which one can use several different orders of a sentence which only slightly changes in meaning. It seems to be reasonable to put more emphasis on the morphological description of pattern elements in terms of scheme introduced in the IPI PAN Corpus of Polish (IPIC) [2]: grammatical class (extended, more fine grained division than Part of Speech) and values of grammatical categories like case, number and gender for nouns and adjectives or aspect and number for verbs. In the case of Polish, the linear positions of LUs in a sentence is not necessarily correlated with their role in a lexical semantic relation when the relation is not symmetrical, e.g. in case of hypernymy most patterns mark hypernym and hyponym by different cases, while their relative positions are changing. Obviously, we can generate many specific patterns for all different combinations, but we can also look for some generalization of a group of patterns on the basis of the morphosyntactic properties.

### B. Scheme of patterns

Patterns have a flat structure and describe a sentence as a sequence of word forms or at most groups of word forms. Patterns are not based on any deeper description of the syntactic structure. The alphabet comprises three types of symbols: an empty symbol `*`, *base form* and *matching place*. The empty symbol represents any LU (represented by any of its word form). The base form is a morphological base form of some LU together with the grammatical class, as the same morphological base form can represent more than one LU. A matching place represents all LUs whose morphosyntactic description matches the partial description encoded in the matching place symbol. As grammatical classes of IPIC are too fine grained we introduced a macro collective symbol, e.g. `noun` joining together: *substantives*, *gerunds*, *foreign nominals* and *depreciative nouns*. A matching place is a reduced version of the IPIC morphosyntactic tag, in which only some grammatical categories are specified.

Following [5], there are always two matching places: one at the beginning and one at the end of a pattern. Patterns do not describe the left and right context of a potential instance.

A pattern also encodes the roles of both LUs identified by matching places, e.g.:

**`(hypo:subst:nom) jest (hyper:subst:inst)`**

—where `jest` is *to be*$_{number=sg,person=3rd}$, `hipo` marks hyponym, and `hiper`—hypernym, `subst`—*substantive*, `nom` and `inst` are case values (all three are from the IPIC descrption)

**(hyper:subst:inst) jest (hypo:subst:nom)**

The described pattern scheme expresses to some extent the characteristic features of Polish. The change of the pattern scheme was the first step leading to an algorithm called Estratto, which is a modification of Espresso that is better suited for an inflectional language like Polish.

## V. Induction of patterns and extraction of instances

According to [5], patterns can be inferred by any pattern learning algorithm. In Estratto the generalisation and unification of patterns is based on the longest common substring algorithm. The algorithm is guided by a predefined list of relation specific LUs, e.g. for hypernymy, *być* (*to be*), *stać się* (*to become*), *taki* (*such*), *inny* (*other*), etc.

In Espresso, the inferred patterns are then generalized by replacing all *terminological expressions* (i.e. a subset of noun phrases) by *terminological labels*. Such an approach to generalization is not applicable for Polish, as a required *chunking parser* (chunker) does not exist. Therefore a slightly different method was proposed. Patterns are grouped and then merged with respect to the significant elements of the patterns: specification of matching places (determining properties of morphological similarity to contexts), and words expected to be related in some way to the semantic relation being extracted.

The instance extraction phase comes after patterns induction and selection. An instance is a pair $\langle x, y\rangle$ of LUs belonging to the set of instances representing the target semantic relation. Authors of Espresso suggest, that if the algorithm is applied to a small corpus, two methods can be used to enrich the instance set. First each multiword LU in an instance can be simplified according to the head of LU. For example *new record of a criminal conviction* is simplified to *new record* and this to *record*. A new instance is created with a simplified LU and the LU that was in the pair together with the original LU. Second an expansion is made by an instantiating pattern only with one of the LUs: $x$ or $y$, and searching if it can extract a new instance from additional corpora, for example, for the instance (*dog*, *animal*) and the pattern expressed in the inflectional format used in Estratto:

**(hypo:subst:nom) is a/an (hyper:subst:inst)**

two queries:

**dog is a/an (hyper:subst:inst)**

and

**(hypo:subst:nom) is a/an animal**

are created.

Instances gathered using both of those methods are added to the instance set. However, it worth of noticing that in all experiments described in [5] only one-word LUs are used and the applied corpora are claimed to be large enough to provide statistical evidence.

Generalized patterns, described above, are not classified as *generic* as long as they do not generate ten times more instances than the average number of instances extracted by specific patterns. However high recall results in loosing some of the precision, that is why every instance extracted by a generic pattern is verified. The verification process starts with instantiating all specific patters with the instance in question. Then the instantiated patterns are queried in a validating corpus and the confidence measure is next computed on the basis of collected frequencies. If confidence is above the defined threshold than the tested instance is considered as representing the target relation.

In Espresso, Internet resources were used as a huge validating corpus for instances extracted by generic patterns. Contrary to this, due to several limitations in searching the Internet in Polish (i.g. limited access to the search engine and inflection of the language), we applied a second large corpus, much smaller the aforementioned one, as a validating corpus in Estratto. The necessary condition is that the validating corpus must be similar in its characteristics to the basic one.

The process of the induction of patterns and extraction of instances is controlled by the following set of parameters:

1) the *number of top* $k$ patterns not to be discarded (preserved for the next iterations),
2) the *threshold* for measure of confidence for instances,
3) the *minimum* and *maximum frequency* values for patterns,
4) the *minimum size* of a pattern— all patterns that consist of only matching places and conjunctions are discarded by the assumption,
5) a *filter* on common words in instances and instances that have identical LUs on both positions,
6) the *size* of the validating corpus.

## VI. Performance measures

A proper evaluation of the extracted lexical semantic resources is mostly a serious problem, e.g. [13], [14]. However, in the case of lists of instances the situation is simpler: we need to verify how many of them are correct. There are only two possibilities to compare the list with: an existing manually constructed resource, i.e. plWordNet in our case or human judgement. The former will introduce some bias as plWordNet is limited in its size, but gives a possibility of testing the whole set of instances, while the manual evaluation is always laborious.

In both types of comparison we applied the standard measures of *precision* and *recall*, e.g. [15]. The F-measure could not be applied, because of the limitations of recall based on plWordnet, which are discussed later.

Precision is defined in a standard way: $P = \frac{tp}{tp+fp}$ where $tp$ is the number of true positives, i.e. extracted pairs of LUs which are instances of the target relation, $fp$—false positives.

True positives are patterns or instances (depending on what we are going to measure) that are correct and marked by algorithms as correct and false positives are those that are incorrect but marked by algorithms as correct.

Recall is defined in a standard way too: $R = \frac{tp}{tp+fn}$

However, it is worthy of note that recall is the ratio between instances or patterns that are correctly marked as correct (true positives) and the sum of true positives and all those that are correct but were either marked as incorrect or not extracted at all. The problem is that we certainly cannot treat the limited plWordNet as the exhaustive description of the subsequent relations. Thus, recall in our approach is only the measure of the ratio of rediscovery the plWordNet structure, it is not a recall in relation to all instances or patterns that can be present in the used corpora.

We extracted a ranked list of possible instances which can be sorted in descending order of their reliability. Its values are real numbers and there is no characteristic point below which we can cut off the rest of pairs according to some analytical properties. Thus, instead of pure precision and recall, we prefer to use *cut off precision* and *cut off recall* calculated only in relation to some $n$ first positions on the sorted list of results (instances or patterns).

Finally, we used the following evaluation measures:

1) *Cut off precision based on plWordNet* - this measure marks as correct only those instances and patterns that were found both in plWordNet and an additional list provided a priori by human judge. It is worth to consider that the limited size of plWordNet can influence precision negatively because some LUs are not present yet or although included into plWordNet, still not connected. This precision is computed for each element on the list of instances.
2) *Precision based on human judgement* is evaluated according to a randomly drawn sample from the list of instances. This evaluation measure was used only for the first group of experiments (ref. Experiments). The error level of sample was 3% and the confidence level was 95%.
3) *Recall based on plWordNet* is evaluated at the set of word pairs generated form plWordNet. However this measure does not describe the recall from corpora

## VII. Experimental setup

The experiments were performed on three datasets corpora:

a) IPIC [2] including about 254 millions of tokens, is not balanced but contains texts of different genres: literature, poetry, newspapers, legal texts and stenographic records from parliament, and scientific texts,

b) 100 millions tokens from Rzeczpospolita [16]—Polish newspaper (henceforth RC)

c) and a corpus of large text documents collected from the Internet, texts including larger numbers of spelling errors and duplicates were semi-automatically filtered out (LC), LC includes about 220 millions of tokens.

We tested several configurations of systems during the experiments, namely:

a) **ESP-** —Espresso without generic patterns,

TABLE I
INFLUENCE OF THE EXTENDED RELIABILITY MEASURE AND CHANGES IN THE FORM OF PATTERNS

| | **Precision levels (plWN)** | Hum. eval | Recall plWN | **Instances** |
|---|---|---|---|---|
| **ESP-** | 36%/50%/75% | 39% | 27% | 3982/903/14 |
| **ESP-nm** | 37%/50%/75% | 47% | 26% | 3784/774/96 |
| **ESP+** | 32%/50%/75% | | 27% | 4221/613/11 |
| **EST-** | 52%/52%/75% | 54% | 18% | 1628/1628/169 |
| **EST-nm** | 16%/50%/75% | 59% | 18% | 1775/1598/120 |

b) **ESP-nm**—Espresso without generic patterns, but with the extended reliability measure 5,

c) **ESP+** Espresso with generic patterns,

d) **EST-** Estratto without generic patterns, exploiting specific features of Polish,

e) **EST-nm** Estratto without generic patterns, exploiting specific features of Polish language and the extended reliability measures 5,

f) **EST+nm** same as VII but using generic patterns.

If not stated otherwise the threshold for confidence is 1.0 for all ESP systems and 2.6 for EST. The *number of top* $k$ patterns was set to $k = 2 + I$, where $I$ is the number of the present iteration. The number of iterations was set to four. In those experiments whose results are presented we focused only on the hypo/hypernymy relation and we selected as the main corpus, on which we performed experiments compared in tables, was IPIC.

## VIII. Experiments

Research on Espresso and Estratto can be divided into three groups. The first one includes experiments, that were designed to analyse the influence of the proposed extended reliability measure 5 and of the form (i.e. if they are improved for using selected aspects of Polish or no) of patterns for the **ESP-**, **ESP-mn** and **EST-**, **EST-nm**. The results are shown in Table I, where values in the column labelled "Precision level" refer to the number of instances in the last column e.g. in the first row **ESP-** extracted 3982 instances with precision of 36%, 903 instances with precision 50% and so on. The column labelled “Hum. eval” refers to the evaluation of the results made by one of the authors.

On the basis of the results of the first group of experiments, Table I, one can conclude, that the use of the original reliability measure 1 results in extraction of more instances. The overall cut-off precision based on plWordNet for 4000 of instances is around 35%. On the other hand, the cut-off precision evaluated for EST suggested, that EST performs worse. However plWN is relatively small, that is why, the evalutation can be misleading. Therefore one of the authors performed manual evaluation. This additional evaluation showed, that in fact the plWN might be used only for a very rough estimation of the precision. The results of the manual evaluation suggest also, that the use of the new measure increases the precision of **ESP-/EST-**. In each case ESP-nm vs. **EST-nm** is better. During experiments it was also observed, that the value of original

TABLE II
DEPENDENCY OF THE ALGORITHMS ON THE VALUES OF PARAMETERS.

| | **Precision levels (plWN)** | Recall plWN | Instances |
|---|---|---|---|
| **EST-nm:th1.0** | | | |
| **EST-nm:th2.6** | 16%/50%/75% | 18% | 1775/1598/120 |
| **EST-nm:th5.2** | **47%/50%/75%** | **20%** | **1907/1736/117** |
| **EST+nm:patt4iter4** | 27%/50%/75% | 27% | 4372/1537/86 |
| **EST+nm:patt8iter4** | 32%/50%/75% | 24% | 3999/1521/47 |
| **EST+nm:patt4iter6** | 17%/50%/75% | 29% | 7265/1505/83 |
| **EST+nm:patt8iter6** | 30%/50%/75% | 27% | 4210/1485/58 |
| **EST+nm:PMI** | **27%/50%/75%** | **27%** | **4187/1505/86** |
| **EST+nm:Tscore** | 6%/- | 25% | 8934/-/- |
| **EST+nm:Zscore** | 34%/50%/75% | 26% | 3563/1419/59 |

reliability (1) decreases very fast and after 6th iteration it is far below 10–12. This is a reason for the drop of newly extracted instances. Applying the extended reliability (5) allows to avoid that problem. Recall based on plWN is comparable, and depends on the number of extracted instances. Another matter of concern is the scheme of the patterns adjusted for Polish. It is clear that the application of the adjusted patterns produces better precision **EST-** and **EST-nm** in comparison to **ESP-** and **ESP-nm**. However the recall is decreased.

Experiments from the second group were performed only for **EST+nm** and **EST-nm**, using suggested measure, and this group was aimed at determining the influence of the algorithm parameters on the result. The following dependencies were investigated:

i) influence of the confidence threshold on the precision of instances achieved within subsequent iterations,

ii) influence of the number of the top $k$ patterns on the stability of the algorithm and the precision of instances,

iii) dependency on the filtering infrequent and very frequent patterns and instances.

iv) influence of the number of initial instances (seeds) on the induced patterns, and then the influence of the ration between instances and patterns inducted by them,

v) a way in which different statistical similarity measures used in reliability calculation change the precision of the results.

In the case of **i)** it seems, that best results are achieved, when the threshold is higher, see Table II. However one must keep a balanced ratio between chosen instances and new patterns. If there is a small number of instances, there is no statistical evidence to induce proper patterns and EST/ESP crawls picking almost random patterns. That leads to the decrease in precision.

Considering **ii)**, on the basis of the obtained numbers, it can be noticed, that using a smaller number of the $k$-top patterns results in higher precision. This is due to the stability of a model, in which semantic relations are generated by a small group of elite patterns. An interesting idea would be to use a dynamic $k$-top factor. Should it be more strict at the begging, the more stable set of patterns would be indicated. Then in the subsequent iterations the $k$-top would grow faster, and as a result more correct patterns could extract instances from corpus in the next phase.

The data for **iii)** are not presented in Table II. However the experiments have shown, that infrequent patterns (occurring less than four times) should be filtered before generalization, because they introduce additional noise, which causes good patterns to be evaluated as worse.

Initial seeds, the point **iv)**, are meant to generate a skeleton of a model of the lexical semantic relation. If the number of seeds is not enough high, the best extracted patterns can be random. Of course, one could collect a small number of seeds, that would indicate only expected patterns. However that would require a precise analysis of the corpus, that would be used for instance extraction. That is pointless, because using more seeds one can acquire the same patterns with less effort.

In the case of **v)**, the data shows, that PMI is better than Z-score and T-score as the measure of similarity in the extraction of lexical semantic relations. T-score results are especially disappointing, and that might be due to the fact of the insufficient statistical evidence (the algorithm very often accepted instances occurring only once).

The third and the last group of experiments was prepared to check the ability of EST and ESP to use a different corpus and extract other relations than hypo/hypernymy. Performed experiments showed that both algorithms: EST and ESP can be applied to different corpora successfully, however it seems, that each time the corpus is changed, a new confidence threshold must be discovered by some method For IPIC the threshold value was 2.6 but in the case of RC we found 0.9 as working fine. Tests performed on the LC corpus appeared to be unsuccessful. But this is a rather special case, as most of the text in LC are written in literary style, so the language expressions are more complex. Moreover, one should expect less defining sentences than in utility texts. It seems that this kind of corpus requires more powerful patterns to catch some syntactic dependencies. The other problem, namely the application of EST to different relation types appeared to be only partially successful. Tests on meronymy ended with a rather poor result, i.e. the estimated precision was lower than 30%. There are at least three main reasons for this failure. Firstly, the expressive power of patterns is too low and some important morpho-syntactic dependencies are missed. Secondly, meronymy is indeed a set of quite varied sub-relations. That is why, it could be reasonable to try to extract each sub-relation separately. Thirdly, the trials were done only on one corpus. On the other hand, initial experiments on extracting antonymy (but only for adjectives) gave promising results. The human-judged cut-off precision reached 39%. Both meronymy and antonymy will be further investigated.

## IX. EXAMPLES

Below we present examples of instances (hyponym; hypernym) extracted by the **ESP-** algorithm from IPIC:

*szkoła*(*school*); *instytucja*(*institution*)
*maszyna*(*machine*); *urządzenie*(*mechanism*)
*wychowawca*(*tutor*); *pracownik*(*employee*)
*kombatant*(*combatant*); *osoba*(*person*)
*bank*(*bank*); *instytucja*(*institution*)
*pociąg*(*train*); *pojazd*(*vehicle*)
*telewizja*(*television*); *medium*(*medium*)
*prasa*(*press*); *medium*(*mass media*)
*szpital*(*hospital*); *placówka*(*establishment*)
*czynsz*(*rent*); *opłata*(*payment*)
*grunt*(*land*); *nieruchomość*(*real estate*)
*Wisła*(*Wisła*); *rzeka*(*river*)
*świadectwo*(diploma); *dokument*(*document*) *opłata*(*payment*); *należność*(*charge*)
*ryba*(*fish*); *zwierzę*(*animal*)
*Włochy*(*Italy*); *kraj*(*country*)
*jezioro*(*lake*); *zbiornik*(*reservoir*)
*jarmark*(*fair*); *impreza* (*entertainment*)
*piwo*(*beer*); *artykuł*(*comestible*)
*zasiłek*(*dole*); *świadczenie*(*welfare, benefit*)
*powódź*(*flood*); *klęska*(*disaster*)
*paszport*(*passport*); *dokument*(*document*)

Examples of patterns extracted by **ESP-** from IPIC and used in the extraction of the above instances are presented below:

```
occ=31 rel=0.26803 (hypo:subst:nom) być
(hyper:subst:inst)
(hypo:subst:nom) is/are (hyper:subst:inst)

occ=20 rel=0.222222 (hypo:subst:nom) i
inny (hyper:subst:nom)
(hypo:subst:nom) and other
(hyper:subst:nom)

occ=26 rel=0.103449 (hypo:subst:inst) a
inny (hyper:base:inst)
(hypo:subst:inst) but other
(hyper:base:inst)

occ=15 rel=0.0684905 (hypo:subst:inst)
przypominać (hyper:subst:acc)
(hypo:subst:inst) resemble
(hyper:subst:acc)

occ=41 rel=0.0263854 (hypo:subst:loc) i
w inny (hper:subst:loc)
(hypo:subst:loc) and in other
(hper:subst:loc)

occ=86 rel=0.00708506 (hypo:subst:nom)
stać się (hyper:subst:inst)
(hypo:subst:nom) become (hyper:subst:inst)

occ=88 rel=0.0060688 (hypo:subst:acc)
interp który być (hyper:subst:inst)
(hypo:subst:acc) interp which is
(hyper:subst:inst)
```

## X. Conclusions and further work

In the paper we presented a partially successful application of the *Espresso* algorithm [5] to Polish. The modified version of the algorithm was called *Estratto*. Experiments showed that the reliability measure proposed by Pantel and Pennacchoti [5] works usually well as a ranking measure for the extraction of lexical semantic relations. However the plWN-based precision of the Espresso/Estratto algorithm is lower when measured on Polish corpora than the precision reported in [5]. This might be due to a slightly different approach to precision evaluation, which was performed partially on the basis of plWordNet (of a limited size) and combined next with a limited manual evaluation. On the other hand the results of the manual evaluation are similar to the results reported in [5]. Results obtained for different measures of similarity as the basis of the reliability suggest that PMI gives the best results for the given test suit.

The adjustment of the pattern structure to the characteristic features of Polish improved the precision in comparison to patterns using only word forms and Parts of Speech as features.

The extended version of Espresso—namely Estratto showed to be successful in extracting hypernyny and antonymy from the IPI PAN Corpus [2] and the Rzeczpospolita corpus [16]. Unfortunately attempts to extract meronymy did not bring positive results.

During experiments we tested several parameters that have a significant influence on the algorithm. The most important of them appeared to be: the *number of seed instances*, the *confidence threshold* and the *number of the k-top patterns* preserved between the subsequent iterations. The number of seed instances should be more than 10. The confidence threshold depends strongly on a corpus, e.g. for IPIC the best found value was about 0.3. Each time the algorithm is applied to a new corpora both seed instances and the measure of confidence must be reset. The number of the $k$-top patterns should be low i.e. about two. Such a number results in a stable representation of the semantic relation, i.e. by the means of the set of patterns. However, it is still unclear, how to explore patterns, that seem to be correct and are close to the top. Those patterns usually disappear in next iterations and that means that some instances are also excluded from final results.

Espresso/Estratto is an intrinsically weakly supervised algorithm, although the preparation of seeds and setting the initial values of parameters might require even some initial runs of Espresso/Estratto or browsing the corpus.

Additionally it turned out, that in order to maintain a stable representation of relations, the appropriate ratio between patterns and instances must be kept. The ratio was estimated during experiments and equals for patterns vs. instances: 1:15/20. If there are less instances, the algorithm becomes unstable. Using more instances results in a longer time of computation.

An interesting result of experiments is the observation of the “intensifying” patterns. Such patterns do not represent any particular semantic relation and when applied alone they extract instances belonging to relations of multiple types. However when the intensifying patterns are combined with regular ones they deliver additional statistical evidence to correct but infrequent instances and as a result rise the precision of the algorithm, e.g.,

**(hypo/holo:subst:nom) w (hyper/mero:subst:inst)**
where *w* means *in*.

We observed a problem with the number of instances collected by the **ESP+**/**EST+** versions of the algorithms. This number is comparable to the number of instances extracted by **ESP-**/**EST-** while one would expect it to be much higher. This might be a result of the characteristic features of the corpus, namely IPIC, used in the experiments or of the size of the validating corpus. This problem might be partially solved by the use of Google as a validating corpus. Unfortunately, in contrast to English, Polish LUs have multiple word forms. As a result queries issued to Google will have to be more complicated. The other reason might be the limited expressive power of patterns. The expressive power of the patterns is the element of the algorithm that should be investigated. The extended structure of patterns still seems to miss some lexico-semantic dependencies, especially in stylistic reach text. The experiments on extracting hypernymy from the corpus LC, mostly consisting of text in literary style, was unsuccessful. The first step towards strengthening patterns is to take into account possible agreements in elements of the patterns that match the instances. The patterns used in EST are very strict about grammatical categories e.g.,

**(hypo:subst:gen) i inny (hyper:subst:gen)**
(two nouns in the genitive case) is treated as a completely different pattern from:

**(hypo:subst:inst) i inny (hyper:subst:inst)**
which matches two nouns in the instrumentative case. It seems to be helpful to allow merging of such patterns into e.g., the form of:

**(hypo:subst:case1) i inny (hyper:subst:case2)**
where case1 = case2. On the basis of the results for **ESP-** and **EST-**, where in **ESP-** there are no such strict constraints, one can expect the increase in recall. The other way, much more complicated, is to enrich the pattern representation, so that additional syntactic information (at least about nominal phrases) could be used.

Natural extension of present representation of instances for Polish is the introduction of multiword lexical units(LUs). Due to their present form it is sometimes possible to obtain instances consisting of two similar words, e.g., for *word office* and *post office* the resulting instance would be (office, office), as only single words are matched.

The list of acquired instances cannot be easily imported to plWordNet. This is due to the fact that the list is a flat structure. Such a representation cannot indicate, which wordnet classes an instance belongs to and what is the distance between the LU in the instance. This problem has been already addressed by Pantel and Pennacchiotti [5].

## ACKNOWLEDGMENT

Work financed by the Polish Ministry of Education and Science, project No. 3 T11C 018 29.

## REFERENCES

[1] V. Mapelli and K. Choukri, "Report on a (minimal) set of LRs to be made available for as many languages as possible, and map of the actual gaps," ENABLER project, Internal report Deliverable 5.1, 2003. [Online]. Available: http://www.elda.org/blark/fichiers/report.doc

[2] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Warsaw: Institute of Computer Science, Polish Academy of Sciences, 2004.

[3] C. Fellbaum, Ed., *WordNet—An Electronic Lexical Database*. The MIT Press, 1998.

[4] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda, "Words, concepts and relations in the construction of Polish WordNet," in *Proceedings of the Global WordNet Conference, Seged, Hungary January 22–25 2008*, A. Tanâcs, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen, Eds. University of Szeged, 2008, pp. 162–177.

[5] P. Pantel and M. Pennacchiotti, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations." ACL, 2006, pp. 113–120. [Online]. Available: http://www.aclweb.org/anthology/P/P06/P06-1015

[6] M. Piasecki, S. Szpakowicz, and B. Broda, "Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns," in *Proc. Text, Speech and Dialog 2007 Conference*, ser. LNAI, vol. 4629. Springer, 2007.

[7] M. P. Bartosz Broda, Magdalena Derwojedowa and S. Szpakowicz, "Corpus-based semantic relatedness for the construction of polish wordnet," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, E. L. R. A. (ELRA), Ed., Marrakech, Morocco, may 2008.

[8] M. A. Hearst, *Automated Discovery of WordNet Relations*. The MIT Press, 1998.

[9] S. Pinker, *Language Learnability and Language Development*. Cambridge, MA: Harvard University Press, 1984.

[10] S. Finch and N. Chater, "Bootstrapping syntactic categories using statistical methods," in *Background and Experiments in Machine Learning of Natural Language*, W. Daelemans and D. Powers, Eds. Tilburg University: Institute for Language Technology and AI, 1992, pp. 229–235.

[11] P. Pantel and D. Ravichandran, "Automatically labeling semantic classes," in *HLT-NAACL 2004: Main Proceedings*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 – May 7 2004, pp. 321–328. [Online]. Available: http://acl.ldc.upenn.edu/N/N04/N04-1041.pdf

[12] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora." in *Proceeedings of COLING-92*. Nantes, France: The Association for Computer Linguistics, 1992, pp. 539–545.

[13] T. Zesch and I. Gurevych, "Automatically creating datasets for measures of semantic relatedness," in *Proceedings of the Workshop on Linguistic Distances*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 16–24. [Online]. Available: http://www.aclweb.org/anthology/W/W06/W06-1104

[14] M. Piasecki, S. Szpakowicz, and B. Broda, "Extended similarity test for the evaluation of semantic similarity functions," in *Proceedings of the 3rd Language and Technology Conference, October 5–7, 2007, Poznań, Poland*, Z. Vetulani, Ed. Poznań: Wydawnictwo Poznańskie Sp. z o.o., 2007, pp. 104–108.

[15] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 2001.

[16] "Korpus rzeczpospolitej," [on-line] http://www.cs.put.poznan.pl/dweiss/rzeczpospolita.

[17] ACL 2006, Ed., *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. The Association for Computer Linguistics, 2006.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Using UML State Diagrams for Visual Modeling of Business Rules

Konrad Kułakowski
Institute of Automatics,
AGH – University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
Email: kkulak@agh.edu.pl

Grzegorz J. Nalepa
Institute of Automatics,
AGH – University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
Institute of Physics, Jan Kochanowski University
ul. Żeromskiego 5, 25-369, Kielce, Poland
Email: gjn@agh.edu.pl

***Abstract*—Recently, in response to growing market demand, several different techniques of business rules representation have been created. Some of them try to present business rules in a visual manner. However, due to the complexity of the problem, the graphic representations that are proposed seem to be far from perfection. In this paper we would like to describe how UML state diagrams might be used for business rules formulation and visual modeling. The strength of this approach relies on reusing classical notions provided by UML 2.0, e.g. an action, guard, etc., in a way which is close to theirs original meaning.**

## I. Introduction

RULES constitute a commonly recognized mechanism for representing knowledge about the world. In particular they are suitable for specifying the behavior and properties of different complex artifacts like information systems [1]. The rule-based approach is a foundation of various engineering and business systems. It is helpful for formulating business knowledge about the problem domain, defining the way in which systems interact with the changing environment and performing inference upon the knowledge. With time, rules applied to business problems have gained the name *business rules* and have become a separate notion. From the very beginning, business rules have aimed to be precise enough for professional software engineers and easy to use and to understand for all parties involved in the modeling of business domain concepts [2]. This is especially important since domain experts usually do not have mathematical knowledge indispensable for using formalisms like Prolog, Datalog or Process Algebras, for example.

The simplicity and expressiveness have also been very important for UML's authors. Since UML is perceived as a universal modeling language, in a natural way there is a tendency to use it for rule modeling [3], [4]. Growing popularity of languages like URML [5] proves that UML has been recognized as a very useful platform for business rules modeling. UML, thanks to being popular with the software and business community, has an emerging opportunity to became an everyday language for wide audience of people involved in various kinds of business activities.

The paper is organized as follows: In Sect. II related research in the area of business rules is discussed. Next, in Sect. III, selected rule modeling aspects are summarized. Then, in Sect. IV a new approach to rule representation with UML is proposed. Finally, in Sect. V, concluding remarks, as well as directions for future work are given.

## II. Related Works

The first known usage of the term "Business Rules" comes from 1984 [6]. In fact, applying rules to business logic started in the late 1980s and the early 1990s[7], [8] and focused mainly on using business rules for data base modeling and programming. A serious attempt to make business rules better defined is "The Business Rules Book" written by Ronald G. Ross [9] and the report of the IBM GUIDE "Business Rules" Project [2]. In these works the authors define the scope of the problem domain, and identify core categories and patterns of business rules.

There is no uniform business rule format [10], [11]; however, there are some standardization efforts in this area [12]. Also the idea of using UML together with business rules is not completely new. Usually, UML is treated as a language for expressing facts about terms in a model [13], whilst the rules themselves are not written in UML. In this context, the applying of UML/MOF to modeling rules, not only to the terms or facts, seems to be a very interesting perspective. There are several projects that try to propose UML/MOF representation for rules. One of them is *Production Rules Representation* (PRR) proposed by OMG [10]. PRR has been developed to address the need for a representation of production rules in UML models (business rules modeling as part of a modeling process). It proposes a meta-model for production rules and defines several notions like condition, action, binding and rulesets. The relationship between PRR and OMG model driven architecture is also discussed.

Another interesting initiative developed in order to exchange rules between communities is a general markup framework for integrity and derivation rules (R2ML) [14], [15]. The authors of R2ML define the rule concepts on the basis of RuleML[11] and Semantic Web Rule Language (SWRL) in terms of MOF and UML[15]. On the top of the list of concepts provided by RuleML [16], a UML-Based Rule Modeling Language

978-83-60810-14-9/08/$25.00 © 2008 IEEE

(URML) has been developed. It extends UML meta-model with the notion of a rule and defines new diagram elements supporting visual notations for rules [3], [5]. In this approach, modeling rules is done with the help of a class diagram enriched by one new diagram element called a *conclusion arrow*. A created model must conform to the *URML* meta-model, defining the semantics for all indispensable notions, i.e. rules, conditions, conclusions, etc.

Besides the indisputable benefits like providing visual rule notation in accordance with *UML/MOF*, the relatively high number of classes required for defining a single rule might be a little onerous for people not accustomed to work with large *UML* models. A new diagram elements such as rule's circle requires use of special *UML* tools supporting *URML* syntax.

Business rules express the statements upon the model elements called business vocabulary. Thus it is important to have well formulated business vocabulary with precisely defined semantics. As an example of a standard facilitating business vocabulary formulation may serve SBVR (Semantics of Business Vocabulary and Business Rules) [17]. SBVR defines the vocabulary and rules, which allow to express business vocabulary, business facts and business rules. This standard also provides XMI scheme for the interchange of created artifacts among different software tools. However, SBVR is not an UML based language, since its generality, it might be successfully used in the context of UML model. In such approach detailed semantics of business vocabulary is defined with the help of SBVR in Structured English, whilst some aspects of business vocabulary are also expressed in form of UML class diagram (e.g. EU-Rent Example [17]).

## III. MODELING BUSINESS RULES

Let us take a closer look at the modeling concept first. The situation is as follows: by having a natural language description of a certain problem area, we aim at providing a declarative rule-based description of this area. The rule-based description is then formalized (or at least disciplined) compared to the original one. Rules are a knowledge representation method that captures regularities, constraints and relations. While formalized, this description is a high-level one, close to the original natural language-based one. So the basic sense of rule modeling is to build a rule-based knowledge representation of the problem. It is a classic case of knowledge engineering, where a designer, knowledge engineer has to identify, extract, describe and represent knowledge possessed by domain experts, or possibly embedded in an information system, such as an enterprise.

The rule representation should meet certain requirements. It should:

- be easy to grasp by non-technical individuals,
- be possible to process automatically and to integrate with a certain runtime (rule engine),
- formalized to some degree,
- meet certain quality standards (e.g. completeness, lack of redundancy),
- be suitable for interchanging and integration with other systems,
- be manageable.

The emphasis on these aspects can differ, depending on the goal of providing the rule-based description. This could be describing system requirements, including constraints, or building a complete system from scratch. Rules can also be thought of as a certain means of formalized communication.

Rule modeling methods and approaches should be considered with respect to other modeling methods such as software engineering methods and methodologies (e.g. UML, MDA). Since rules are often an essential part of business systems, business process modeling methods, such as workflows or BPMN, have to be taken into consideration in the chapter.

When it comes to the modeling *process*, different aspects can be pointed out:

- identifying concepts and their semantics,
- determining high-level structure ruleflow, rulebase contexts,
- building rules capturing the knowledge,
- integrating the ruleset,
- analyzing the quality of the model.

A rule-based model representation is expressed by means of a certain rule language.

While modeling rules, some other important factors have to be taken into consideration. These include:

- rule applications and types, e.g. constraint handling, facts, derivation, etc., and
- rule inference model, mainly the forward and backward chaining case.

These issues can have an important influence on the rule language.

### A. MODELING LANGUAGES

Rule modeling is a classic problem in the field of AI (Artificial Intelligence). It is a question of knowledge engineering (KE) and building rule-based expert systems that have strong logical foundations. In this chapter, some fundamental logical rule formats are considered, based upon the propositional or predicate calculus. The formats are a basis for rule languages. Rules can be practically written and processed in the logic programming paradigm, e.g. in Prolog. Even though the language uses a subset of first order predicate logic (restricted to Horn clauses), it is easy to write meta-interpreters working with languages of another order.

Within the AI, a number of *visual* knowledge representation methods for rules have been considered. These methods include:

- decision tables, that help combining rules working in the same context,
- decision trees, that support visualization of the decision making process, and

- decision graphs and lists, a less common but powerful method of control specification.

Two important factors for using these methods are:

1) design support – all of these methods help the designer (knowledge engineer) develop the rule-based model in a more rapid, and scalable manner, and
2) logical equivalence – all of these formally correspond to rules on the logical level.

These methods are used to model rules in practical applications. They also influenced some classic software engineering languages, e.g. UML.

A common approach to model rule-based systems is to use UML, considered by some as a universal modeling language. UML offers a visual or semi-visual method for different aspects of information modeling. By using this, it is possible to model some specific rule types. However, when it comes to practical knowledge engineering, it has some major limitations due to the different semantics of rules and the object-oriented paradigm. In particular cases, some of these shortcomings can be overcome by the use of OCL, which allows for constraint specification for UML classes.

One area where UML or UML-related methods are more useful is the conceptual modeling, which supports practical rule authoring. UML class diagrams are suitable to capture relations between concepts present in rule vocabularies. In this context usage of SBVR from OMG seems to be very interesting.

Since UML is a de facto standard information modeling method in software engineering approaches and tools, it can be treated as a low-level language on top of which a richer semantics is provided. This is possible for the standardized MOF and UML profiles formats. By building upon these, a dedicated rule modeling language can be built, e.g. URML or PRR.

An important community is built around the W3C and the so-called *Semantic Web Initiative*. The methods built on top of XML, RDF and OWL allow also for rule modeling for both web and general purposes. Rule interchange is possible using the XML-based RIF format.

The rule-based model can be used as a stand-alone logical core of a business application. However, in practice, this model should be somehow integrated with other models, and components of a heterogeneous application. Examples of integration discussed in this chapter include integration with business processes described with BPMN, as well as interfaces on the Java platforms. A number of approaches to the integration can be enumerated, with Model-View-Controller being a prime example.

Finally, the multilayer aspect of the rule language should be considered. A useful and expressive rule language should provide:

- rich, but well-defined semantics,
- formally defined syntax with clear logical interpretation,
- scalable visual representation, which allows for the visualization of many rules,
- machine readable encoding for model interchange and integration.

Using this criteria, it is easier to analyze selected languages.

### B. HEKATE APPROACH

Developing new effective rule methods is one of the main goals of the HEKATE project [18]. It aims at providing a complete rule modeling and implementation solution. Some of the main concepts behind it are:

- providing an integrated design and implementation process, thus
- closing the semantic gap, and
- automating the implementation, providing
- an executable solution, which includes
- an on-line formal analysis of the design, during the design.

To fulfill the goal HeKatE uses methods and tools in the areas of:

- knowledge representation, for visual design,
- knowledge translation, for automated implementation,
- knowledge analysis, for formal verification.

Currently, development within the project is focused on the:

- conceptual design method, ARD+ [19], which allows for attribute (vocabulary) specification,
- logical design, XTT+ [20], for rule design using a hybrid decision tables and tree based method.

For project progress see hekate.ia.agh.edu.pl

A principal idea in this approach is to model, represent, and store the logic behind the software (sometimes referred to as business logic) using advanced knowledge representation methods taken from KE. The logic is then encoded with the use of a declarative representation. The logic core would be then embedded into a business application or embedded control system. The remaining parts of the business or control applications, such as interfaces or presentation aspects, would be developed with classic object-oriented or procedural programming languages such as Java or C.

The work proposed in this paper aims at developing a UML-based representation for rules describing the logical core of an application. Such a representation would allow for direct rule modeling with standard UML tools.

## IV. REPRESENTING BUSINESS RULES IN UML

Rules are widely recognized as a critical technology for building various types of knowledge-based applications. Rules are also important in information systems engineering, where they constitute a natural way of expressing business application logic. The classical form of a rule is a plain, textual if-then-else statement defining a rule's condition and rule's conclusions. On the other hand, there are a few propositions of visual rules modeling, e.g. URML [3].

In response to the market gap for visual rules modeling and taking into account the great popularity of UML as a general purpose modeling language, we propose another UML-based approach to visual rules modeling. In this approach a rule is

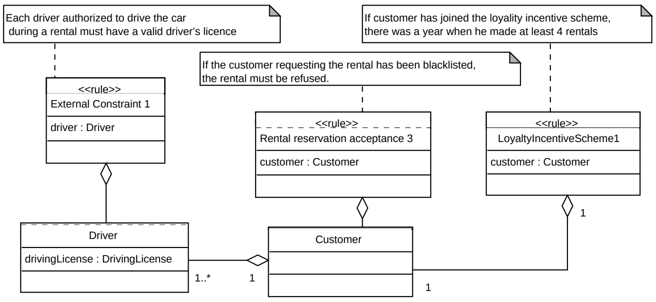

Each driver authorized to drive the car
during a rental must have a valid driver's licence
If customer has joined the loyality incentive scheme,
there was a year when he made at least 4 rentals
If the customer requesting the rental has been blacklisted,
the rental must be refused.
<<rule>>
External Constraint 1
driver : Driver
<<rule>>
Rental reservation acceptance 3
customer : Customer
<<rule>>
LoyaltyIncentiveScheme1
customer : Customer
Driver
drivingLicense : DrivingLicense
Customer
1..*
1
1
1

Fig. 1. Rules as classes

expressed as a class with a stereotype *rule*. Such a class has its own *state diagram*, which is used for expressing the rule's condition, conclusion and action.

Presented solution does not deal with business vocabulary modeling. It is assumed that all the business artifacts indispensable to business rules formulation are given in form of class diagrams. The precise definition of business vocabulary might be provided in form of other modeling languages e.g. SBVR.

### A. INTERPRETABLE LANGUAGE

In the presented approach, UML is used for modeling a system and creating schemes of the rules. Other important parts of the model, such as events, guards conditions, actions and conclusions, are written in an *interpretable language*. In the context of UML the natural choice of language for expressing e.g. guard conditions is OCL. It has appropriate expression power and proven syntax constructions suitable for expressing statements upon the UML abstracts. Since actions are also important parts of some kinds of rules, there is still a need for another language for actions modeling. OCL as a constraint language does not seem to be the optimal choice for this purpose. On the other hand, a situation in which there are several different languages for modeling several different aspects of the systems is not convenient. The optimal solution should consists of UML and one interpretable language having mechanisms allowing for the expression of all non-UML terms like constraints and actions. Such a language should be easy to use by rule architects, and it also needs to be understood by the rule interpreter. Because people for whom the idea of using UML and state diagrams appears attractive should not be forced to use a certain implementation of a rule engine, the question of an interpretable language to expressing non-UML model elements remains open. Thus, however, the OCL language seems to be useful in the context of specifying logical statements upon the UML terms, in this article, it is treated rather like one possible and well documented proposition, not like a part of the final method's specification.

### B. RULE DIAGRAMS

The rule diagram (Figure 1) is a UML class diagram containing the classes representing rules and some business terms that are directly used by the rules.

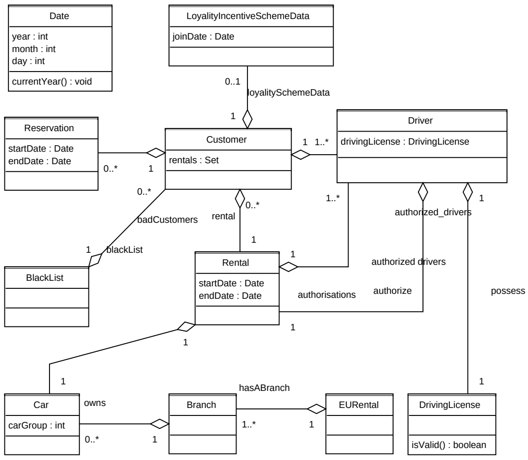


Fig. 2. Business vocabulary diagram

The main role of the rule diagram is to show relationships between rules and business terms. Attributes of the class representing a rule should cover all of the business artifacts indispensable for a condition evaluation, a conclusion drawing or an action being performed. Every rule should have a textual comment informing about its business source.

### C. BUSINESS VOCABULARY DIAGRAMS

Rules express some logical statements about terms and facts [2] that comes from the UML model. The set of all terms and facts will be called the *business vocabulary*. Thus, every UML diagram containing elements of *business vocabulary* which are neither *rule diagrams* nor *rule definition diagrams* associated with a rule will be called a *business vocabulary diagram*. The most popular kind of *business vocabulary diagram* is a class diagram (Figure 2).

### D. RULE DEFINITION DIAGRAM

The definition of the rule has a form of a state diagram associated with the class representing the rule. This diagram will be called the *rule definition diagram*. The rule can be fired (can be applied) if, according to its *rule definition diagram*, it is able to change the state from a start state to a stop state. If there are some actions defined between a start state and a stop state, all of them have to be executed when the rule is triggered. If it is not possible for the rule to leave a start state, it means that the given rule is not active and cannot be executed at the moment. Following the conceptual rule classification [15], [3] we would like to define three types of business rules: Integrity Business Rules, Derivation Business Rules and Reactive Business Rules. The Reactive Business Rules incorporates Production Business Rules and ECA Business Rules [21].

The integrity rule, represented here as an integrity constraint, consists of a predicate function given producing a boolean value on the output. A very popular language for formulating constraints and expressing business knowledge in UML models is OCL (Object Constraint Language). It allows for precise formulation properties of relations between classes

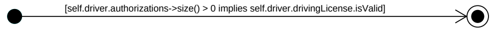

[self.driver.authorizations->size() > 0 implies self.driver.drivingLicense.isValid]

Fig. 3. Integrity business rule

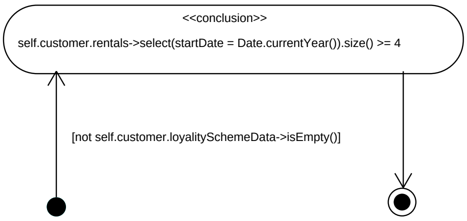

<<conclusion>>
self.customer.rentals->select(startDate = Date.currentYear()).size() >= 4
[not self.customer.loyalitySchemeData->isEmpty()]

Fig. 4. Derivation business rule

and objects. In fact, a guard condition may be formulated in any language understood by a rule interpreter; however, for the sake of examples' clarity, the OCL language is preferred. In the proposed approach the semantics of an integrity rule is given by a simple start-stop diagram containing one guard condition. Obviously, a condition of the integrity rule is met if the rule object changes its state to stop. On figure 3 an example of integrity rule is shown. The integrity constraint expresses the fact that the driver's driving license is valid if it has at least one authorization. The source of this rule as well as other presented examples comes from broadly known in literature as the EU-Rent case study [17], [15], [3] .

Derivation business rules have conditions and conclusions. Depending on the positive evaluation of the condition, a conclusion is drawn. In our approach, a condition is represented by a guard expression, whilst the conclusion is the action performed in the action state followed by a guard expression. The action state should have a stereotype *conclusion*. The action should have the form of a logical expression in a language understood by a rules interpreter. The action's logical expression represents a new knowledge derived from the existing facts (subjects of conditions) in the system. The presented example (figure 4) shows derivation rule describing the fact that if a customer has joined the loyalty incentive scheme, he must have made four rentals within the year. The same as previously the source of the rule is the EU-Rent case study.

Reactive business rules may have conditions, triggering events and actions. For the given rule, one condition or one triggering event (at least one is obligatory) and one action should be defined. In general, such a kind of rule allows for the modeling of an event-condition-action behavioral pattern, in which execution of the action is preceded by event triggering and guard condition evaluation. The absence of a condition is allowed only if a triggering event is defined, and inversely, a triggering event is not required if only a condition is defined. Thus, there are three possible subtypes of the reactive business rules:

- reactive business rule with a non-empty event and a non-empty condition
- reactive business rule with a non-empty event and an empty condition
- reactive business rule with an empty event and a non-empty condition

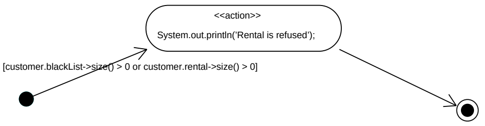

<<action>>
System.out.println('Rental is refused');
[customer.blackList->size() > 0 or customer.rental->size() > 0]

Fig. 5. Reactive business rule

The first kind of rule is triggered by the event only if a guard condition is true. The second one models the executing action in response to the raised event, whilst the third kind of rule models the action execution as a result of changes in the system that make the guard condition true. The condition is modeled as a guard expression; thus, it may have any boolean form suitable for a rule interpreter (OCL is the preferred language). The action state in the *rule definition diagram* has a stereotype *action*. Whilst the first two kinds of reactive rules have a semantics of ECA Rules, the third kind of the reactive rules has a semantics borrowed from production rules i.e. an action is executed as soon as the condition becomes true [21]. On figure 5 an example of the third kind of reactive business rule is shown. According to the rule, if a customer is on a black list or he has already a rental, the specified action is executed. In this particular case the action writes the message informing that the rental is (or will be) refused.

## V. CONCLUSIONS

In this paper, the main assumptions of a new approach to representing business rules in UML has been presented. This approach allows for modeling business rules as UML state diagrams. It makes modeling rules similar to modeling system behavior, which may shorten the time required for modeling the system. A rule is represented by a well-known concept of a stereotyped class; thus, there is no need to define any new UML artifacts except for stereotypes. Consequently, almost every UML 2.0 compatible modeler might be used for rule modeling. It is easy to find the business vocabulary since it is explicitly shown in UML diagrams. With the help of stereotyped rules, the well known statechart concepts, such as action, guard and event, retain as much as possible from their original meaning. E.g. since applying the rule is represented by following the transitions of an state diagram, a guard concept remains a kind of expression deciding whether we may apply the rule, i.e. whether we may follow the transition.

Since some of a rule's components are written in OCL or other interpretable languages, rule modeling may seem to be a little bit harder than using a graphical notation. On the other hand, such languages are usually quite simple; e.g. in OCL some more sophisticated constructions like nested collections has been abandoned [22]. Thus, after getting a bit of practice in the chosen language, working with rules written in UML and state diagrams should not be a problem.

Regardless of the lack of a strictly defined language for actions and guards expressions, some experiments in these areas are being conducted. The aim of the authors is to propose a complete *xtUML* solution [23], which would allow to execute rule–based model on appropriate rule–based runtime engine as well as provide model building guidelines to facilitate modeling process. Since key role of UML statecharts in existing *xtUML* solutions [24], [25] a statechart form of a rule cannot be underestimated. A semi–automatic transition from *SBVR* textual form to *business vocabulary diagram* and *rules diagram* is also considered. Using *SBVR* would allow for easy capturing business vocabulary and business rules, and theirs validation and preliminary authorization. In the context of *MDA* [26] such transition will correspond to transformation a computation independent model *(CIM)* to platform independent model (*PIM*). The next transition, i.e. from *PIM* to *PSM*, will be done by rule–based runtime engine.

The work presented here will be integrated within the *HeKatE* approach briefly discussed in Sect. III-B. The basic idea is to model a rule-based logical application core with the visual representation presented here. The OCL expression can be replaced by Prolog-based rules, since Prolog is the language of choice for the HeKatE prototype implementation [27]. Since HeKatE aims at designing applications using the Model-View-Controller pattern, using an UML-based representation greatly improves the possibility of integration with the UML-based view design. Another area of intensive research is the formal analysis of the rule-based model. It is hoped that HeKatE verification methods could be extended to cover the UML-based model.

**Acknowledgements** The paper is supported by the HeKatE Project funded from 2007–2009 resources for science as a research project.

## References

[1] S. Russell and N. P., *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[2] D. Hay and K. A. Healy, "Defining business rules what are they really?" the Business Rules Group, Tech. Rep., 2000. [Online]. Available: http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf

[3] S. Lukichev and G. Wagner, "Visual rules modeling," in *Ershov Memorial Conference*, ser. Lecture Notes in Computer Science, I. Virbitskaite and A. Voronkov, Eds., vol. 4378. Springer, 2006, pp. 467–473. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-70881-0_42

[4] ——, "UML-Based Rule Modeling with Fujaba," 2006. [Online]. Available: http://oxygen.informatik.tu-cottbus.de/i1papers/LukichevWagnerFujabaDevDays2006.pdf

[5] G. Wagner, A. Giurca, and S. Lukichev, "Modeling Web Services with URML," in *Proceedings of Workshop Semantics for Business Process Management 2006, Budva, Montenegro (11th June 2006)*, 2006. [Online]. Available: http://idefix.pms.ifi.lmu.de:8080/rewerse/index.html

[6] D. S. Appleton, "Business rules: the missing link," *Datamation, 15*, vol. 30, no. 16, Oct. 1984.

[7] R. Ross, "Entity modelling: Techniques and application," Database Research Group, Boston, MA, Tech. Rep., 1987.

[8] C. C. Fleming and B. von Halle, *Handbook of Relational Database Design*. Reading: Addison-Wesley Professional, 1989.

[9] R. G. Ross, *The Business Rule Book*. Business Rule Solutions, 1994.

[10] S. Tabet, G. Wagner, S. Spreeuwenberg, P. D. Vincent, J. Gonzaques, M. C. de Sainte, J. Pellant, J. Frank, and J. Durand, "OMG production rule representation - context and current status," in *Rule Languages for Interoperability*. W3C, 2005. [Online]. Available: http://www.w3.org/2004/12/rules-ws/paper/53

[11] H. Boley, "The ruleML family of web rule languages," in *PPSWR*, ser. Lecture Notes in Computer Science, J. J. Alferes, J. Bailey, W. May, and U. Schwertel, Eds., vol. 4187. Springer, 2006, pp. 1–17. [Online]. Available: http://dx.doi.org/10.1007/11853107_1

[12] H. Boley and M. Kifer, "RIF basic logic dialect," World Wide Web Consortium, Working Draft WD-rif-bld-20071030, Oct. 2007.

[13] T. Halpin, "Verbalizing business rules: Part 1," Apr. 03 2004.

[14] G. Wagner, "How to design a general rule markup language," Jun. 2002, invited talk at the Workshop XML Technologien für das Semantic Web (XSW 2002), Berlin. [Online]. Available: citeseer.ist.psu.edu/wagner02how.html

[15] G. Wagner, A. Giurca, and S. Lukichev, "A general markup framework for integrity and derivation rules," in *Principles and Practices of Semantic Web Reasoning*, ser. Dagstuhl Seminar Proceedings, F. Bry, F. Fages, M. Marchiori, and H.-J. Ohlbach, Eds., no. 05371. Internationales Begegnungs und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. [Online]. Available: http://fparreiras/papers/R2ML.pdf

[16] G. Wagner, G. Antoniou, S. Tabet, and H. Boley, "The abstract syntax of ruleML - towards a general web rule language framework," in *Web Intelligence*. IEEE Computer Society, 2004, pp. 628–631. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/WI.2004.134

[17] D. Chapin, "Semantics of business vocabulary and business rules (SBVR)," in *Rule Languages for Interoperability*. W3C, 2005. [Online]. Available: http://www.w3.org/2004/12/rules-ws/paper/85

[18] G. J. Nalepa and I. Wojnicki, "A proposal of hybrid knowledge engineering and refinement approach," in *FLAIRS-20 : Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference : Key West, Florida, May 7-9, 2007*, D. C. Wilson, G. C. J. Sutcliffe, and FLAIRS, Eds., Florida Artificial Intelligence Research Society. Menlo Park, California: AAAI Press, may 2007, pp. 542–547.

[19] ——, "Towards formalization of ARD+ conceptual design and refinement method," in *FLAIRS2008*, 2008, submitted.

[20] ——, "Proposal of visual generalized rule programming model for Prolog," in *17th International conference on Applications of declarative programming and knowledge management (INAP 2007) and 21st Workshop on (Constraint) Logic Programming (WLP 2007) : Wurzburg, Germany, October 4–6, 2007 : proceedings : Technical Report 434*, D. Seipel and et al., Eds. Bayerische Julius-Maximilians-Universitat Wurzburg. Institut für Informatik, september 2007, pp. 195–204.

[21] B. Berstel, P. Bonnard, F. Bry, M. Eckert, and P.-L. Patranjan, "Reactive rules on the web," in *Reasoning Web*, ser. Lecture Notes in Computer Science, G. Antoniou, U. Aßmann, C. Baroglio, S. Decker, N. Henze, P.-L. Patranjan, and R. Tolksdorf, Eds., vol. 4636. Springer, 2007, pp. 183–239. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74615-7_3

[22] J. Warmer and A. Kleppe, *The Object Constraint Language: Precise Modelling with UML*, ser. Object Technology Series. Reading/MA: Addison-Wesley, 1999.

[23] S. Flint and C. Boughton, "Executable/translatable UML and systems engineering," in *Practical Approaches for Complex Systems (SETE 2003)*, 2003.

[24] M. Kostrzewa and K. Kułakowski, "A practical approach to the modelling, visualising and executing of reactive systems," in *MIXed DESign of integrated circuits and systems*, 2006, pp. 705–710.

[25] S. Burmester, H. Giese, M. Hirsch, D. Schilling, and M. Tichy, "The fujaba real-time tool suite: model-driven development of safety-critical, real-time systems," in *27th International Conference on Software Engineering (ICSE 2005), 15-21 May 2005, St. Louis, Missouri, USA*, G.-C. Roman, W. G. Griswold, and B. Nuseibeh, Eds. ACM, 2005, pp. 670–671. [Online]. Available: http://doi.acm.org/10.1145/1062455.1062601

[26] A. Kleppe, J. Warmer, and W. Bast, *MDA Explained. The Model Driven Architecture: Practice and Promise*. Addison-Wesley, 2003.

[27] G. J. Nalepa and A. Ligęza, "Prolog-based analysis of tabular rule-based systems with the xtt approach," in *FLAIRS 2006 : proceedings of the nineteenth international Florida Artificial Intelligence Research Society conference : [Melbourne Beach, Florida, May 11–13, 2006]*, G. C. J. Sutcliffe and R. G. Goebel, Eds., Florida Artificial Intelligence Research Society. FLAIRS. - Menlo Park: AAAI Press, 2006, pp. 426–431.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 195 – 200

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Discovery of Technical Analysis Patterns

Urszula Markowska-Kaczmar
Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27,
50-370 Wrocław, Poland
Email: Urszula.Markowska-Kaczmar@pwr.wroc.pl

Maciej Dziedzic
Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27,
50-370 Wrocław, Poland
Email: 133644@student.pwr.wroc.pl

***Abstract*—In this paper our method of discovering data sequences in the time series is presented. Two major approaches to this topic are considered. The first one, when we need to judge whether a given series is similar to any of the known patterns and the second one when there is a necessity to find how many times within long series a defined pattern occurs. In both cases the main problem is to recognize pattern occurrence(s), but the distinction is essential because of the time frame within which identification process is carried on. The proposed method is based on the usage of multilayered feed-forward neural network. Effectiveness of the method is tested in the domain of financial analysis but its adaptation to almost any kind of sequences data can be done easily.**

## I. Introduction

THE issue of discovering data sequences has been heavily investigated by the scientists of different disciplines for many years. Despite this fact there is no doubt the issue is still up-to-date. Statisticians, economists, weather forecasters, operating system administrators – all of them, in their daily routine, deal with many kind of sequences. Specifically, in the domain of finance analysis there are patterns defined by the *Technical Analysis* (TA). Recognition of some of this patterns among quotation data triggers investors buy or sell decisions regarding examined stock. So it is crucial for the people who play the stock exchange to recognize patterns when they are really formed by stock exchange quotations. Because of that there is a need to provide trustworthy method of finding defined sequences. Lately, discovery of patterns in time series plays very important role in the area of bioinformatics [2] also.

In this paper a method of discovering data sequences in the domain of financial analysis is presented but its adaptation to any other kind of sequences data can be easily done. This method uses multilayered feed-forward neural network to recognize the technical analysis patterns. All experiments which aim was evaluation of the method efficiency, are done by the use of data which come from the Warsaw Stock Market.

The paper consists of five sections. The next one describes different approaches to the problem of sequence data discovery. Our method is introduced in the third section. The next one presents the results of the experiments. some of them were performed by the use of the method in artificial environment simulating the Warsaw Stock Market. The final section presents conclusion and future plans.

## II. Related Works

Methods of pattern discovery in time series sequences in the financial analysis are closely connected to econometrics which can shortly be defined as the branch of economy that deals with defining models of different systems by the use of mathematics and statistics. Some of these models are created by economists in order to make analysis of data or to make a prediction of future stock exchange quotations. The problem is to prepare a good model, where ‘good’ means the model which takes into consideration all important relations which can be distinguished in the modeled reality. This is of course not easy. Often some relations become important under certain circumstances when others turn out to be useless. To comply with all defined requirements there is a need to prepare accurate model which can consist of even hundreds of equations. Such approach causes difficulties in its comprehensibility by the user but also in a computer implementation. That is why scientists look for other methods of discovering patterns in time series.

Fu and others [3] describe a method which uses *perceptually important points* (PIPs) of the graph to compare it with other graph. By PIPs are assumed points that are significant for the shape of the diagram to which they belong. Authors presented a method for finding PIPs and algorithms for determining the distance between points from two different graphs. The idea introduced by them reflects the human-like way of thinking (people usually do not remember all the points which build the graph – they keep just more significant ones in mind and then compare them to the other important points). The advantage of this algorithm is its easy implementation. Despite of that fact, there is a big disadvantage of this  method of discovering sequences. A problem is with series which have high amplitude between two adjacent points – higher than some PIPs can place between those two points. It leads to the problem, when we have PIPs identified not among whole series but mainly in some its parts. Similar approach is applied in the paper [1], where a special metrics of similarity between a pattern in question and a given pattern is designed.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

The usage of rules and fuzzy rules in searching time sequence patterns are considered as well. The examples can be found in [7] and [3].

Many researches are made by the use of machine learning methods in order to retrieve some predefined technical analysis patterns within the time series, e.g. [5].

Very popular approach is an application of Kohonen's neural network to cluster patterns retrieved among stock exchange quotations. The examples of SOM networks can be found in [4] and [6]. Authors admitted that this kind of network in their experiments showed good results in searching for patterns of main trend of quotations. They also consider this approach as not ideal for making predictions of turning points among quotations.

Other approach which used neural networks is presented in [5]. The method described in this paper can be shortly characterized as follows. Each of the patterns is memorized as a chart in the computers memory within some specified boundaries. Next, neural network (NN) is trained of chosen pattern. After training, the network is able to recognize whether a given series is similar to the pattern it was trained. To become results more trustworthy the author suggested to use two different NNs for a recognition of one pattern (the average of both results was treated as a final result). What is important, both neural networks had to be trained using different sets of learning patterns. The method based on chart pattern recognition in time sequence is proposed in  as well.

## III. The Details of the Method

Our method of discovering data sequences in time series is also based on the neural network which has feedback connections. It is trained with back propagation learning algorithm. The whole idea is simple. For each pattern of technical analysis one dedicated neural network exists which is trained to recognize it. The architecture of the network used in the experiments is presented in Fig 1. The network is fully connected. Each of the inputs represents exactly one value of a stock exchange quotation. In this figure *N* describes the number of input neurons (which was set to 27 in the experiments), *L* represents the number of hidden neurons (it was equal to 14 in the experiments) and *M* is the number of output neurons (that was set to 1). The response of the output neuron indicates whether a given series is recognized as a pattern that the network was trained to recognize.

To be more precise it is worth mentioning, that sigmoidal function was used as an activation function. This means that the value returned by the output neuron is in the range (0; 1).

The output value closer to the upper bound of the range was interpreted as a given series was similar to the series from training set. When the continuous range of values is allowed the obvious question is how to make a binary decision if the series represents a pattern in question or not? The answer is not so well-defined. It depends on what the parameters of the network training were set, what the stop criteria of learning algorithm were adjusted or what kind of activation function was chosen. In the experiments after preliminary experiments this threshold value was set to 0.85. In the Fig. 2 there are presented main steps of the proposed method of discovering data sequences. In the first step training patterns for neural network are prepared. It is important to provide representative patterns. It is a good practice that some of them should be multiplied within the training set (with added noise).

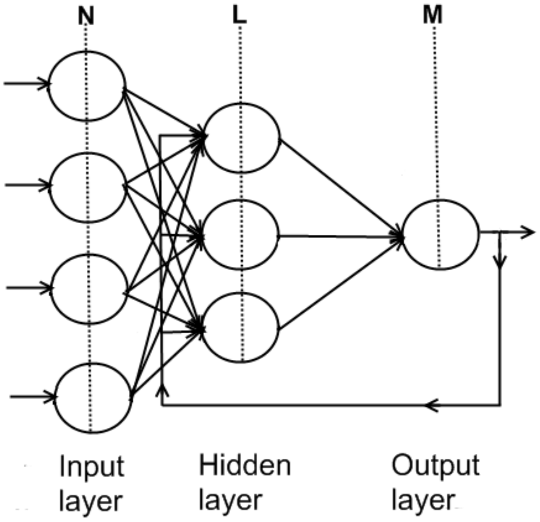


Fig. 1. The neural network architecture used in the experiments

```
PrepareTrainingPatterns()//define training set
NormalizePatterns()//prepare normalization
TrainNeuralNetwork()//training process
SmoothInputSeries()//step is optional
NormalizeInputSeries()//series' normalization
ProceedtheSeries()//Classifying decision
```

Fig. 2. The algorithm of discovering technical analysis patterns in time series

Adding similar learning patterns ensures that the neural network after the training process will have better generalization skills. The next step (normalization of patterns) is needed in order to reduce all defined patterns to a common range. It is important, because in other case series defined on different ranges could favor some patterns with higher values. Each value *s* from a series *S* is normalized according to the equation (1). In the next step the neural network is trained. The training process should be continued until an output value of the network reaches satisfied value (usually below defined threshold).

$$s_{norm,i} = \frac{s_i - \min(S)}{\max(S) - \min(S)}, \qquad (1)$$

where: $s_i \in S$, $\min(S)$ – minimum, $\max(S)$ – maximum.

In the next step a given series, in order to be processed by the neural network, can be smoothed. It is especially essential when a series consists of any abnormal values. The aim of smoothing is to reduce the number of points where the amplitude between two adjacent points in the chart is extremely high. An example of smoothing result is presented in Fig. 3. Because this method changes the original points in a chart it is recommended to use it only if needed.

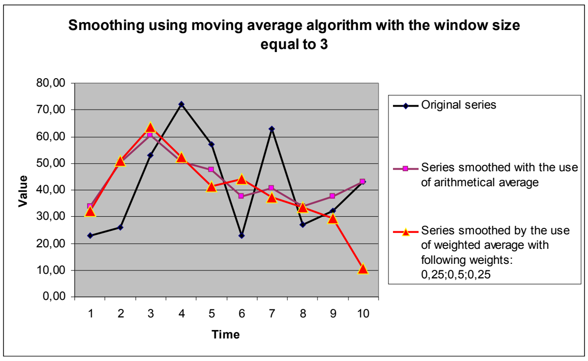


Fig. 3. An example of smoothed series

Because of the fact that patterns used during a training process were normalized, similar action has to be done after an optional smoothing of the series. It is crucial to have series defined on the same range as training patterns are. Otherwise the result cannot be reliable. The final step relies on processing values from the series given to the input layer of the network and calculating its output.

The algorithm shown in Fig. 2 can be easily used when the series (*P*) in question is of the same length as the patterns from the training set (series *S*).

The problem raises when the lengths are different. Having patterns longer than a series in a training set leads to the necessity of expanding a given series (i.e. by adding some additional points between existing ones). Depending on the shape of series which needs to be stretched, different methods should be used. A simple approach can be realized by the use of linear function to calculate values of extra points while more complex can demand the usage of more sophisticated curves in order to determine points values (such as Bezier curves).

The other case that should be considered is when a given series *P* is longer than the number of inputs in the neural network (a length of training patterns). Then a shortening of examined series should be done. In this case to solve this problem the following solutions can be suggested:

a) Shortening by a deletion of surplus points,

b) Shortening by a determining only perceptually important points (based on the idea presented in [3]),

c) Shortening by compressing a series.

The first technique is based on the assumption that some points from a time series can be removed without affecting its shape too much, which is especially true if concerned are series taken from the real stock exchange. In this case each subseries formed as hop or valley on the chart consists of many points which values change gradually. Removing one point from such short subseries will not affect a whole form. The simplest way to determine which points should be removed is to count how many of them is surplus ($s_p$). Afterwards the number of all points ($m$) in the series should be divided by $s_p$ resulting in the steps ($k$) which should be used while designating indexes of surplus points within a given series. Fig. 5 presents the effect of the usage of the mentioned method to the series which is depicted in the Fig. 4.

The second technique is to find within a series exactly $n$ characteristic points (called perceptually important points -*PIP*). Other points, which were not considered as characteristic points should be removed.

The last technique of shortening series of length $m$ to become one with $n$ values is its compressing. The compression can be done by specifying $n$ segments in a given series and all values within each segment are substituted by one value. This value is an arithmetic or a weighted average of the substituted values (this method is a little similar to smoothing). The exemplary results are shown in Fig. 6.

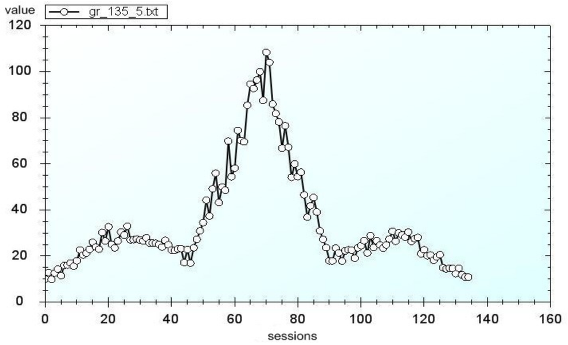


Fig. 4. The chart of 'Head and shoulder' pattern made of 135 points

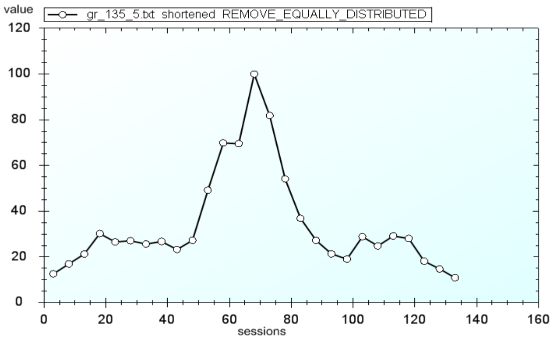


Fig. 5. The chart of 'Head and shoulder' pattern shortened to 27 points using a deletion of surplus points

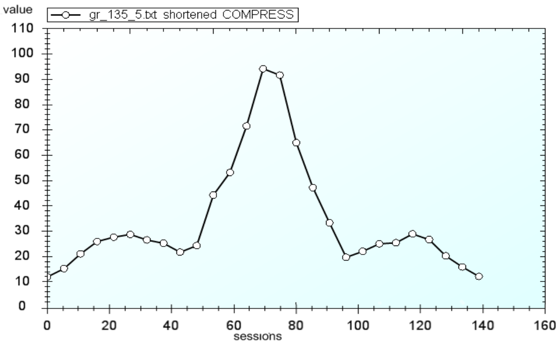


Fig. 6. The chart 'Head and shoulder' pattern shortened to 27 points using a compression

It is important to emphasize that all described previously issues were adequate to recognize a whole series as a pattern. The other case is when we want to find how many times an interesting pattern was repeated among the whole series (this operation can only be done, when a series is longer then training patterns). One approach to this problem is to specify start index and the number which represents a value of length step (which will be used for moving the window from the start index). Next, moving a window (which has the length equal to the number of input neurons of the neural network) the main series can be cut into subseries with a defined step from the start index. Then, each subseries should be checked whether it is similar to the pattern trained by the network. The problem becomes more complex when the length of subseries differs from the length of training patterns. We can consider checking the subseries of length from 2 up to *m* (where *m* represents the length of whole series).

In this case the problem occurs that computational complexity becomes $O(m^2)$ . To reduce the number of subseries that should be checked, similarly to [3] a function *TC* (given by eq. (2)) is used. Its task is to control a length of a series which should be processed. This function returns a smaller value when the length of the series is closer to the preferred length. In eq. (2) *dlen* is the desired length of series (which in our case should be equal to the number of input neurons in the network), *slen* means the series length. Additionally, *dlc* parameter can be adjusted according to the steepy of the function which is used. Only for the points which are below specified threshold (i.e. $\lambda = 0.2$ ) on the *TC* function graph the checking should be performed.

$$TC(slen, dlen) = 1 - \exp^{-(d_1/\theta_1)^2}, \qquad (2)$$

where $d_1 = slen - dlen$ , $\theta_1 = dlen/dlc$

Fig. 7 illustrates how on the basis of *TC* function the lengths of subseries are checked. The following values of parameters were used: $dlen = 180$ , $dlc = 2$ , as *slen* all values of series were provided. For assumed value $\lambda = 0.2$ red bolded line marks the range of the lengths to be checked.

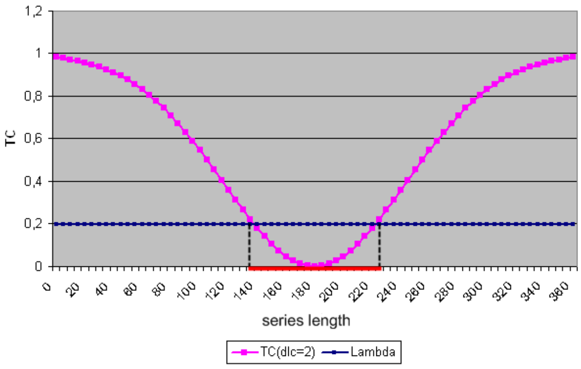


Fig. 7. Example of *TC* function usage

## IV. Experimental Results

In the first experiment the shortening methods of series were evaluated in order to choose the best one. The network was trained to recognize the technical analysis pattern 'head and shoulder'. The training set was prepared where each training pattern had the length 27 (the number of neural network inputs). It consists of positive training patterns (representing 'head and shoulder' form) as well as negative ones (that do not represent this form). The network was trained with an error equal to 0.001.

To evaluate the method of a shortening series the testing set was created. It contained: 30 artificial series of 'head and shoulder' pattern with the length equal to 54; 81 and 135 (10 of each length), 10 series of 'triple top', 'double top' and some randomly chosen patterns, finally some series of archive stock (GPW) exchange quotations (which were manually annotated by the authors whether they represent the pattern in question - 'head and shoulder' form or not. In these annotations the value 1 informs that a given time series represents a given pattern, value 0 means that it does not).

In the test it was arbitrary assumed that the network output equal or greater than 0.9 represents the neural network recognition of the pattern in question.

For each pattern from the testing set an *error* between desired value of the output and the one returned by the network was used to evaluate the results (absolute value of subtraction of mentioned elements). The average error calculated for each method is the basis of comparison. The results are shown in Table I.

Table I
Comparison of shortening techniques

| Shortening technique | Average error | Deviation of average error |
|---|---|---|
| **Surplus points** | 0.0672 | 0.1233 |
| **Compression** | 0.0697 | 0.1409 |
| **PIP** | 0.0936 | 0.1632 |

It is easy to notice that the best results are achieved by the method *surplus points*. The results in Table II show that it performs discoveries of patterns in the best way, as well. Effectiveness of patterns discovering was calculated as a relative number of properly recognized patterns to the number of all patterns.

Table II.
Effectiveness of discovering patterns using different shortening techniques

| Shortening technique | Effectiveness | | |
|---|---|---|---|
| | artificial series | GPW series | All series |
| **Surplus points** | 1.0000 | 0.8730 | 0.9624 |
| **Compression** | 1.0000 | 0.8571 | 0.9577 |
| **PIP** | 0.8400 | 0.8413 | 0.8404 |

Based of an analysis of the results we can draw the conclusion that the proposed method of checking whether a given long series (longer than the number of inputs in the network) is similar to a chosen pattern returns very good outcome. For all presented techniques of shortening we can observe that effectiveness is greater than 80%, considering two best techniques we received even better result (~95% of properly classified series).

The aim of the next experiment was to check whether the methods of discovering patterns are sensitive to the length of tested subseries. For the test purpose one long series was chosen. It was created on the basis of stock exchange quotations of the stock market 01NFI from 150 sessions (from 14 August 2006 till 16 March 2007). The algorithm of discovering patterns has run twice. In the first run the length of the window varied from 2 to 100. In the second one the *TC* function was applied. It allowed to limit the number of time sequence lengths to be checked (the range from 22 to 32 for the network with 27 input neurons). The results are presented in [8]. The blue line represents the widths of window for which patterns could be found in the given series without using *TC* function, while the pink line shows the number of discovered patterns with the use of *TC* function. It can be easily notice that its usage really limits the range of widths to the (21; 32). The fact, that for the widths of window in the range 60 – 69 so high number of patterns were found can be a surprise. But it is nothing extraordinary. We have to keep in mind, that the neural network with well performed preprocessing algorithm (which properly shortens or expands series) can effectively recognize patterns regardless of the length of checked series. The obtained results show that the method is not very sensitive to the length of the tested time series.

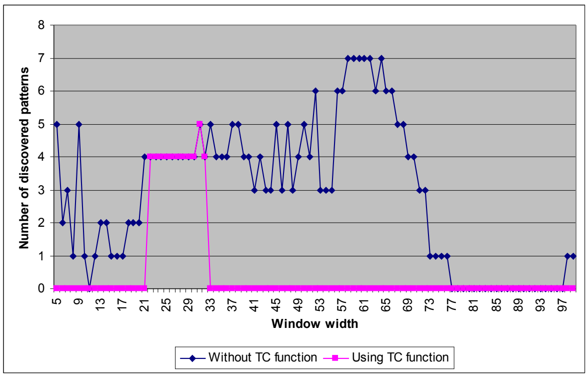


Fig. 8. The number of discovered patterns in relation to the window width

In Fig. 9 and Fig 10 examples of the series found during the experiment are presented (the red line represents a shape of the chart ‘head and shoulder’).

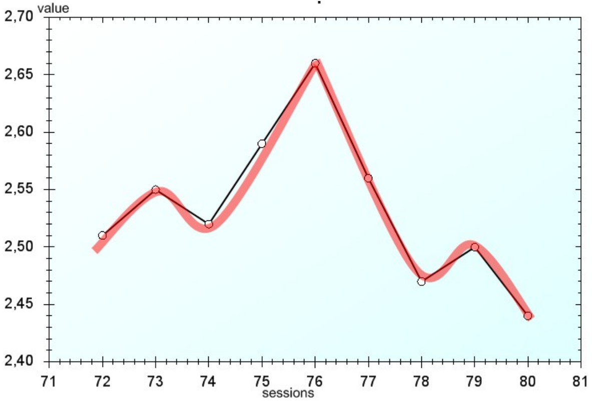


Fig. 9. Stock exchange quotations of 01NFI formed from 9 sessions identified as a 'head and shoulder' pattern

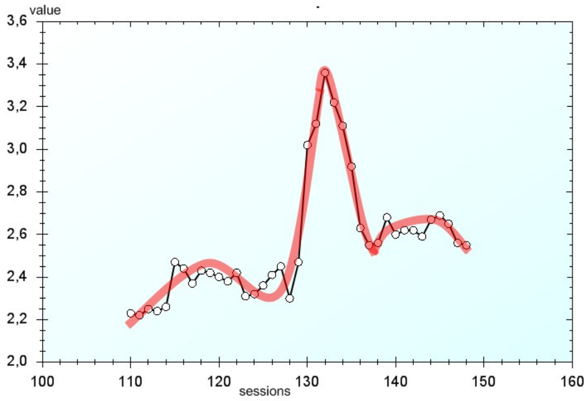


Fig 10. Stock exchange quotations of 01NFI formed from 39 sessions identified as a 'head and shoulder' pattern

As it was mentioned before, the method of discovering data sequences in a time series was tested also in the artificial environment – multi-agent stock exchange system which is presented in . In this system agents representing real investors are evolved by genetic algorithm. Each agent is described by the set of its coefficients defining its behavior. The aim of the system is to find the set of agents (with the best suited values of coefficients) who will be able to generate the stock price movement similar to the existing one in the real stock. Evolution takes place in steps which are called generations. After each generation the individuals (the set of agents in our case) are evaluated in terms of fitness value that informs about a quality of an individual. The better is the fitness value, the better is the set of agents (individual). Originally the system had an naïve algorithm (assigned as *old*) of identification which investments should be done by an agent. Then this algorithm was substituted by the method of discovering time series sequences presented in this paper (called *new*). The comparison of the results with the usage of both methods is shown in Table III.

An analysis of the results in the table clearly shows that the application of the new method improves the value of agents’ fitness. The old algorithm returned good results only in the third test. It means that stock prices generated by the agents using newer decision algorithm are much more similar to the real ones. However, because a genetic algorithm has embedded randomness in its nature, more tests are required to fully evaluate the results which were not possible to perform now because of the duration of one experiment.

TABLE III.
THE COMPARISON OF NEW AND OLD ALGORITHM OF TAKING DECISION BY THE AGENTS

| | **Average fitness of all individuals in all generations** | | **Fitness of the best individual in the experiment** | |
|---|---|---|---|---|
| **Nr** | Decision algorithm | | | |
| | *old* | *new* | *old* | *new* |
| 1 | 0.12 | 0.25 | 0.47 | 0.51 |
| 2 | -0.01 | 0.19 | 0.43 | 0.45 |
| 3 | 0.11 | 0.05 | 0.59 | 0.50 |
| 4 | 0.02 | 0.03 | 0.42 | 0.42 |

It is worth mentioning that the platform on which tests were performed should be upgraded in some places (i.e.

agents should start with an amount of money adequate to the number of stocks which are on the market, genetic algorithm should not create a specified number of new agents as the result of mutation operator after generation, etc.). For the purpose of this test no upgrades were performed (only the mentioned change of the decision algorithm took place). Authors suspect, that even better results could be gained by the use of newer discovering patterns method if some patches to the existing platform were provided. Performed experiment was a first trial of integration and has shown that there is still some place for improvements.

## V. Conclusion and Future Plans

The aim of the research presented in this paper was to design of an effective method which is able to properly recognize a given pattern in the time series data. Based on the results of experiments we can draw the conclusion that the proposed method can properly discover the sequences of data within time series. Moreover, when the network is trained the process of recognition is easy and fast. The network response arrives immediately. The only difficulty can be the network training – the choice of appropriate training patterns and the parameters of training, but after some trials and getting more experience this problem disappears.

However the results are promising, there are still improvements possible, for instance other optimization technique of finding the series of shorter or longer widths than the number of input neurons in the network could be proposed. As it was shown *TC* function limits the number of searched widths but it is not the ideal solution, because some proper patterns can be omitted. Some improvements can be made in the test platform, as well. Some upgrades in this system can have an impact on the trustworthy of the performed tests. All mentioned problems and places where improvements can be made are great opportunity to continue studies on the proposed discovering technical analysis pattern method.

## References

[1] Fanzi Z., Zhengding Q., Dongsheng L. and Jianhai Y., *Shape-based time series similarity measure and pattern discovery algorithm*, Journal of Electronics (China) vol. 22, no 2 , Springer, 2005.

[2] Fogel G. B., *Computational intelligence approaches for pattern discovery in biological systems,* Briefings in Bioinformatics 9(4), pp. 307-316, 2008.

[3] Fu T., Chung F., Luk R., Ng Ch., *Stock time series pattern matching: Template-based vs. Rule-based approaches,* Engineering Applications of Artificial Intelligence**,** vol. 20, Issue 3, pp. 347-364, 2007.

[4] Guimarães G., *Temporal Knowledge Discovery with Self-Organizing Neural Networks. IJCSS*, 1(1), pp.5-16, 2000.

[5] Kwaśnicka *H.* and Ciosmak M., *Intelligent Techniques in Stock Analysis*, Proceedings of Intelligent Information Systems, pp. 195-208, Springer, 2001.

[6] Lee Ch.-H.; Liu A., Chen W.-S., *Pattern discovery of fuzzy time series for financial prediction***,** IEEE Transactions on Knowledge and Data Engineering, vol. 18, Issue 5, pp.613 – 625, 2006.

[7] S., Lu L., Liao G., and Xuan J.: *Pattern Discovery from Time Series Using Growing Hierarchical Self-Organizing Map*, Neural Information Processing, LNCS, Springer, 2007.

[8] Markowska-Kaczmar U., Kwasnicka H., Szczepkowski M., *Genetic Algorithm as a Tool for Stock Market Modelling* , ICAISC, Zakopane, 2008.

[9] Suh S. C., Li D. and Gao J., *A novel chart pattern recognition approach: A case study on cup with handle* , Proc of Artificial Neural Network in Engineering Conf, St. Louis, Missouri, 2004.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Hybrid Differential Evolution Algorithm to Solve a Real-World Frequency Assignment Problem

Marisa da Silva Maximiano
Polytechnic Institute of Leiria. School of Technology and Management. Leiria, Portugal
marisa.maximiano@estg.ipleiria.pt

Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, Juan M. Sánchez-Pérez
Univ. Extremadura. Dept. Technologies of Computers and Communications,Escuela Politécnica. Campus Universitario s/n. 10071. Cáceres, Spain
{mavega, jangomez, sanperez}@unex.es

***Abstract*—The Frequency Assignment is a very important task in the planning of the GSM networks, and it still continues to be a critical task for current (and future) mobile communication operators. In this work we present a hybrid Differential Evolution (DE) algorithm to solve a real-world instance of the Frequency Assignment problem. We present a detailed explanation about the hybridization method applied to DE in order to make it more efficient. The results that are shown use accurate interference information. That information was also adopted by other researchers and it represents a real GSM network, granting, therefore, an extremely important applicability. Furthermore, we have analyzed and compared our approach with other algorithms, obtaining good results.**

## I. Introduction

THE Frequency Assignment problem (FAP) represents an important task in the GSM networks *(Global System for Mobile)*. These networks are very used in the telecommunication area (by mid 2006 GSM services were used by more than 1.8 billion subscribers across 210 countries, representing approximately 77% of the world cellular market) [1].

The FAP problem is an NP-hard problem, therefore approaches using metaheuristic algorithms [2] have proven that they are a viable choice in its resolution.

The main goal of this problem is to be able to obtain an efficient use of the scarcely available radio spectrum on a network. The available frequency band is assigned into channels (or frequencies) which have to be allocated to the transceivers (TRXs) installed in each base station of the network. Both these components have an important role in the definition of this problem. This work is focused on the concepts and models used by the current GSM frequency planning [3]. Moreover, we adopted a developed formulation that takes advantage of realistic and accurate interference information from a real-world GSM network [4].

Our approach presented here uses the Differential Evolution (DE) algorithm as the fundamental step to solve this specific problem. Also, this algorithm w as hybridized with a Local Search method, as well as other features. Therefore, several modifications have been proposed to the original DE to improve its performance further. Also, as mentioned previously to this specific problem a Local Search (LS) method was added in order to be possible to obtain better results to solve this specific real-world FAP problem. It was also applied a method to guarantee that all obtained solutions do not have the most severe penalty, which influences the frequencies assignation inside a same sector of the network.

Our study was focused in a real-world instance of a GSM network named Denver which represents the city of Denver in the USA. It uses 711 sectors with 2612 transceivers (TRXs) installed. For each TRX the same 18 channels are available, representing the available frequencies for the assignation. As we have said, both these elements are the key elements of the problem, influencing directly the codification of the problem and the evaluation of the quality of the solutions accomplished. In a GSM network the TRXs give support to communications, making the conversion between the digital traffic data on the network side and radio communication between the mobile terminal and the GSM network [5].

This paper is structured as follows. In the next section we provide some details about the frequency assignment problem in the GSM networks. Section III describes the algorithm proposed and the hybridization methods applied to it. The results of the experiments are analyzed in Section IV. Finally, conclusions are discussed in the last section.

## II. Frequency Assignment in GSM Networks

In the following subsections we will first give a brief description about the elements present in a GSM network that are fundamental to the Frequency Assignment problem (FAP). Finally it will be presented the mathematical formulation followed by this work.

### *A. The FAP Problem within a GSM System*

A GSM network has several components, but the most important ones, to understand the frequency planning, are the antennas, which are more known as base station transceivers (BTSs) and the transceivers (or TRXs). Therefore, transceivers are the main element to be considered. A BTS can be viewed as a set of TRXs, which are organized in sectors. The

978-83-60810-14-9/08/$25.00 © 2008 IEEE

TRX is the physical equipment responsible for providing the communications between the mobile terminal and the GSM network.

The frequency assignment problem arises because the number of available frequencies (or channels) to be assigned to each TRX is very scarce. Therefore, the available frequencies need to be reused by many transceivers of the network [3][6]. The reuse of the frequencies can compromise the quality of the service of the network. Hence, it is extremely important to make an adequate reuse of the frequencies to the several TRXs, in such a way that the total sum of the interferences occurring in the network needs to be minimized.

Consequently, it becomes extremely important to quantify the interferences provoked by an assignation of a frequency to a TRX and its influence on the remaining TRXs of the other sectors of the network. To quantify this value an *interference matrix* is used, denoted *M* [7]. Each element *M(i,j)* of *M* represents the degradation of the network quality if sector *i* and *j* operate with the same frequency value. This represents the *co-channel* interferences. It also need to be considered the *adjacent-channel* interferences, which occurs when two TRXs, in two different sectors, operate on adjacent channels (i.e., when one TRX operates on channel *f* and the other on channel *f+1* or *f–1*). Therefore, the interference matrix assumes an import role in the formulation of the FAP problem, which intents to minimize the sum of interferences occurring in the network. Thus, it is only possible by using the interference matrix to compute the cost function (see Eq. 1).

### B. Mathematical Formulation

Let $T = \{t_1, t_2, ..., t_n\}$ be a set of *n* transceivers (TRXs), and let $F_i = \{f_{i1}, \dots, f_{ik}\} \subset N$ be the set of valid frequencies that can be assigned to a transceiver $t_i \in T, i=1,\dots,n$ (the cardinality of $F_i$ could be different to each TRX). Furthermore, let $S = \{S_1, S_2, ..., S_m\}$ be a set of given *sectors* (or cells) of cardinality *m*. Each transceiver $t_i \in T$ is installed in exactly one of the *m* sectors and is denoted as $s(t_i) \in S.$ It is also necessary the interference matrix, *M*, defined as: $M = \{(\mu_{ij}, \sigma_{ij})\}_{mxm}$. The two elements $\mu_{ij}$ and $\sigma_{ij}$ of a matrix entry $M(i,j) = (\mu_{ij}, \sigma_{ij})$ are numerical values and they represent the mean and standard deviation respectively, of a Gaussian probability distribution used to quantify the interferences on the GSM network when sector *i* and *j* operate on a same frequency. Therefore, the higher the mean value is, the lower interferences are, and thus it will have a superior communication quality.

A solution to the problem lies in assigning to all the TRXs $(t_i)$ a valid frequency from its domain $(F_i)$, in order to minimize the following cost function:

$$C(p) = \sum_{t \in T} \sum_{u \in T, u \neq t} C_{sig}(p, t, u) \qquad (1)$$

where $C_{sig}$ will compute the *co-channel* interferences $(C_{co})$ and the *adjacent-channel* interferences $(C_{adj})$ for all sector *t* and *u*, in which the transceivers *t* and *u* are installed, that is, *s(t)* and *s(u)*, respectively. $p \in F_1 x F2 x \dots x F_n$ denotes a solution (or frequency plan), where $p(t_i) \in F_i$ is the frequency assigned to the transceiver $t_i$. Moreover, $\mu_{s_t s_u}$ and $\sigma_{s_t s_u}$ are the interference matrix values at the entry $M(s_t,s_u)$ for the sectors $s_t$ and $s_u$ . In order to obtain the $C_{sig}$ cost from equation 1, the following conditions are considered:

$$\begin{cases} k & if\ s_t = s_u, |p(t) - p(u)| < 2 \\ c_{co}(\mu_{s_t s_u}, \sigma_{s_t s_u}) & if\ s_t \neq s_u, \mu_{s_t s_u} > 0, |p(t) - p(u)| = 0 \\ c_{adj}(\mu_{s_t s_u}, \sigma_{s_t s_u}) & if\ s_t \neq s_u, \mu_{s_t s_u} > 0, |p(t) - p(u)| = 1 \\ 0 & otherwise \end{cases} \qquad (2)$$

where *K* is a very large value, defined in the configuration files of the network. The *K* value makes it undesirable to allocate the same or adjacent frequencies to TRXs that are installed in the same sector. In our approach to solve this problem, this restriction was incorporated in the creation of the new solution (frequency plan) produced by the DE algorithm. Therefore, we assure that the solution does not have this severe penalty, which causes the most undesirable interferences (see section III).

## III. Differential Evolution Algorithm

This section is devoted to present the DE algorithm and also its hybridization made by applying a Local Search (LS) method. We also explain the optimization made in the assignment of frequencies to TRXs that are inside the same sector.

### A. Original DE Algorithm

Differential Evolution is an Evolutionary Algorithm proposed by Storn and Price, and has been used successfully to solve optimization problems [8]. The DE has three major steps which are executed in each generation. They are the generation, evaluation and selection, each one involving different operations. These three steps are performed until a stop criterion is not reached. In DE each individual of the current population (named $x_{target}$ or $x_i$) will compete with a new individual $(x_{trial})$ generated by a mutation factor. An individual represents a solution, and it is encoded as a vector of integer values (of frequencies in our case). The DE starts with the creation of an initial population, normally at random.

New individuals are created applying the mutation and the recombination operators. At each generation, for each individual of the population is created a new solution $(x_{trial})$, using a weighted vector difference between two other individuals, selected randomly from the current population. This new individual is obtained e.g. using the equation 3, whose DE scheme name is DE/rand/1/$\beta$, where $\beta$ represents the crossover scheme that can be binomial *(bin)* or exponential *(exp)*. Different classifications of DE schemes are also available [9]. The available schemes use the notation DE/α/γ/β. The α represents the way in which individuals (necessary to obtain the new weighted differences vector, that is, the *trial* individual) are selected from the current population. They can be selected randomly *(rand)* or as the best individual from the current population *(best)*. The γ is the value of difference vector pairs used, which normally are 1 or 2.

$$x_{trial}=x^{t}_{r3}+F\left(x^{t}_{r1}-x^{t}_{r2}\right) \qquad (3)$$

Equation 3 is used to create a new solution, where *r1≠r2≠r3≠i* are used as indices to index each parent vector. F value is the scaling factor, which controls the differential variation (mutation).

Besides the F parameter, it is also necessary the crossover parameter (CR). This last parameter represents a probability that influences the generation of the *trial* individual, by controlling the amount of genes which will be changed from the *target* individual to the new one (after apply the equation 3). It is also necessary to guaranty that all the genes changed will have a value inside the permitted limits. This corresponds to a very important step, in order to guaranty the creation of a valid solution. This characteristic was implemented by using a list of frequency values, containing all the ones that are not been used. Therefore they are available to be assigned to other TRX inside the same sector (each sector of this network problem has one list). Therefore, every time a TRX value is changed, its new value, which is generated by the DE (line 10, Fig. 2), is selected from that list containing the frequencies. This way we guaranty that the values are always inside the valid boundary values.

Table I.
Formulation for the different schemes of the DE algorithm

| Sheme Name | DE mutation definition |
|---|---|
| DE/Rand/1/*β* | $x^{t}_{i}=x^{t}_{r3}+F\left(x^{t}_{r1}-x^{t}_{r2}\right)$ |
| DE/Best/1/*β* | $x^{t}_{i}=x^{t}_{best}+F\left(x^{t}_{r1}-x^{t}_{r2}\right)$ |
| DE/Rand/2/*β* | $x^{t}_{i}=x^{t}_{r5}+F\left(x^{t}_{r1}+x^{t}_{r2}-x^{t}_{r3}-x^{t}_{r4}\right)$ |
| DE/Best/2/*β* | $x^{t}_{i}=x^{t}_{best}+F\left(x^{t}_{r1}+x^{t}_{r2}-x^{t}_{r3}-x^{t}_{r4}\right)$ |
| DE/RandToBest/1 | $x^{t}_{i}=x^{t}_{r3}+F\left(x^{t}_{best}-x^{t}_{r3}\right)+F\left(x^{t}_{r1}-x^{t}_{r2}\right)$ |

The creation of the trial individual is the main characteristic of DE. As we have mentioned previously, several schemas are available [8][9]. The main difference between them is the fact that: they can use different vectors pairs; and also that they could use only randomly-selected individuals or choose to use the best individual obtained so far. Table 1 summarizes the several available schemes, where *t* represents the *t*<sup>th</sup> generation and *r1, r2,* etc. are randomly selected individuals from the population. Steps from line 6 to line 14 (see Fig. 2) may vary depending on the scheme which is being used.

### *B. Hybrid Characteristics*

In order to optimize the results (the frequency plan) two methods have been implemented on the original DE. The first optimization consisted in guarantying that for all the TRXs inside the same sector will not be assigned the same or an adjacent frequency value, because it will provoke severe penalties – these co-channel and adjacent-channel interferences are the highest-cost interferences, as shown in Eq. 2). The second method consisted in applying a Local Search (LS) method, which was adapted to this FAP problem [10]. As described above, this method permits to optimize the assignment of frequencies to every TRX in each sector of a given solution. Also, a more detailed explanation about the LS applied to the FAP problem can be consulted in the reference [10].

The first optimization is used in the creation of the initial population (line 2, Fig. 2) and every time a new solution is created. That occurs every time some of the genes of the solution are changed (line 10 and line 11, Fig. 2). We have optimized this implementation through the incorporation, for each solution, of a dynamic list containing all the available frequencies in each sector. Also, in the LS method we guaranty that the same frequency value or an adjacent frequency is not assigned to two TRXs within the same sector. Therefore, any time a change is made in a solution it is assured that the co-channel and the adjacent-channel interferences are not present. This requires that every time there is made a change to the frequency assigned to a TRX, the list of available frequencies to which belongs the TRX need to be updated (each sector of the network has is one list of available frequencies). Assigning a frequency value *f* to a TRX, originates that also the *f-1* and *f+1* values will be eliminated from the list of available frequencies in the sector of the TRX. To the reverse process (release a frequency value from a TRX) it is necessary to verify that no others TRXs in the same sector are using the *f+1* and *f-1* frequencies values.

Using this approach it is no longer necessary to verify, for each sector, which are the frequencies that are not been used. All that information is always updated in the list of the available frequencies of that sector. Using this approach it was possible to make the process much more efficient.

The second optimization consists in apply the LS method to a solution (frequency plan). Therefore, in a previous work we have conducted several experiments in order to identify what is the most efficient way to apply the LS in the DE [11]. Two hypotheses were considered. The first one consisted in applying the LS to all individuals in the initial population (line 2, Fig. 2) and also after creating every $x_{trial}$ individual (apply the DE scheme, line 10, Fig. 2). The second approach consisted in only applying the LS method after creating every $x_{trial}$ individual.

Due to the experiments carried out the selected strategy was the second approach, because it was the one that permitted obtain the most successfully results, that is solutions with a lower cost [11]. To accomplish these results, and as mentioned previous, several runs using the same DE parameters, were performed in order to identify the location to apply the LS method into the DE algorithm. This experiments were made using a common set of configuration parameters of DE (NP=10; CR=0.4 and F=0.5 and the scheme *DE/Rand/2/Bin*).

```
generation t = 0
population $P_t$ = createInitialPopulation(NP)
evaluate($P_t$)
while (not reached stop criterion)
   for $i$ = 0 to NP do
      select randomly $r1 \neq r2 \neq r3 \neq i$
      $x_{target} = x_i$
      for $j = 1$ to D do //dimension of the problem
         if ($randj[0,1] < CR$ or $j = j_{rand}$) then
            $x_{trial,j} = x_{r3,j} + F\,(x_{r1,j} - x_{r2,j})$          //DE scheme
            $x_{trial,j}$ = applyLocalSearch( $x_{trial,j}$ )
         else
            $x_{trial,j} = x_{i,j}$
         end if
      end for
      if ($f(x_{trial}) < f(x_i)$) then
         $x_i = x_{trial}$
      else
         $x_i = x_i$
      end if
   end for
   t = t + 1
end while
```

Fig 1.DE algorithm with the LS method, using the DE/Rand/1/Bin schema. NP, CR and F are user-defined parameters. NP is the population size.

### C. Solution Encoding

The solution encoding carried out to this problem incorporates the characteristics of this instance of the FAP problem. Therefore, a solution has a dimension equal to the total number of TRXs of the GSM network instance used (in this case 2612, representing the number of TRXs from the network). For every TRX, a frequency (also named channel) has to be assigned. Each TRX has a set of available frequencies. Hence, a solution plan is encoded as a list of integers values *p,* where *p(t)* is the frequency value assigned to TRX *t*.

Each time the DE algorithm changes the value assigned to a TRX, it ensures that the new value is inside the set of the available frequencies for that specific TRX. It also assures that the new value is not assigned to other TRXs installed inside the same sector. According to the made experiments, with this approach it was possible to achieve better results because it is always available an updated list of the unused frequencies.

## IV. Experiments

Our algorithm was implemented in C# using also the Microsoft .NET Framework 3.5. Results were obtained on a PC with a Pentium-4 CPU at 3.2GHz and with 2 GB of RAM, running Windows XP.

The experiments were developed over a real-world instance, named Denver, which has 711 sectors with 2612 TRXs to be assigned a frequency. Each TRX has 18 available channels (from 134 to 151). We only use this dataset in order to be able to compare our results with the results already accomplished for this problem. At this moment is only available this single instance. In future work, when available more real-world instances of the problem we intent to use them.

Fig. 2 displays the network topology, and every triangle represents a sectorized antenna in which operate several TRXs. The interference matrix is the one also used by [4] [10].

All the available information about this instance problem is available in separated configuration files.

### A. Parameterization

Several user-defined parameters are necessary and in the following we present the results achieved with experiments that were performed in order to identify the best configuration parameters to be used by DE. In a previous work [11], we have identify what were the best parameters to be used bt the DE algorithm. The parameters used are the population size (NP), crossover probability (CR), mutation factor/weighting factor (F) and the the DE strategy/scheme. Also, we have determined the best way to apply the local search method in the DE.

The best parameter settings used are a population size (NP) of 10 individuals, a crossover probability (CR) of 0.2, a mutation factor (F) of 0.1, the DE/Rand/2/bin scheme and applying the LS method after the creation of every $x_{trial}$ individual.

We specified as stopping criterion a 30 minutes of execution. In order to provide the results with statistical confidence, we have considered 30 independent runs for each experiment. For each run, and also at every 2 minutes, we have obtained the average, best and standard deviation values.

### B. Results

In order to provide results to be compared with other authors, we have focused on three different time limits (120, 600 and 1800 seconds). Therefore, it is possible to compare different algorithms within short and long time ranges.

According to previous results [10] the more efficient way to create the initial population is randomly but removing the most costly interferences (co-channel and adjacent-channel) within the same sector. After proving the efficiency of this approach to the DE, we used it to initiate the DE algorithm. Initially, without using this approach the accomplished results were not so satisfactory.

After tuning the DE parameters [11] and incorporating also some optimized methods, the conducted experiments showed that using the Local Search method within the algorithm has a very important role in its performance.

According to the formulation presented in section II, the results are given by using the cost function, meaning that a frequency plan with a small value cost is a better frequency plan.

Our results were obtained with the hybrid version of the DE algorithm, which has been described in section III. The best result obtained was 87845.9 cost units (after 30 independent runs, each one with 30 minutes).

Table 2 shows a resume of the results accomplished for this problem by other algorithms with similar characteristics (implemented by other authors), and also our DE algorithm. Here, we show the Ant Colony Optimization algorithm (ACO) and the Scatter Search (SS) [10]. Both have already

TABLE II.
EMPIRICAL RESULTS OF THE METAHEURISTICS FOR 3 DIFFERENT TIME LIMITS. IT SHOWS THE BEST, AVERAGE AND STANDARD DEVIATION OF 30 EXECUTIONS

| Time Limit | 120 seconds | | | 600 seconds | | | 1800 seconds | | |
|---|---|---|---|---|---|---|---|---|---|
| | Best | Average | Std. | Best | Average | Std. | Best | Average | Std. |
| **ACO** | **90736.3** | **93439.5** | 1318.9 | 89946.3 | 92325.4 | 1092.8 | 89305.9 | 90649.9 | 727.5 |
| **SS** | 91216.7 | 94199.6 | 1172.3 | 91069.8 | 93953.9 | 1178.6 | 91069.8 | 93820.4 | 1192.3 |
| **DE** | 92145.8 | 95414.2 | 1080.4 | **89386.4** | **90587.2** | 682.3 | **87845.9** | **89116.8** | 563.8 |

been tested to solve the FAP problem, and they have shown a good performance for this problem.

Analyzing these results, we can conclude that DE obtains the best results for 600 and 1800 seconds, although it is not the algorithm with the best start. Furthermore, with our approach it was also possible to improve the results obtained in [4][12] (where other metaheuristics were used: (1,10) Evolutionary Algorithm, PBIL –Population Based Incremental Learning-, …). In conclusion, the results are quite better that others already achieved for this instance of the problem.

## V. CONCLUSION

In this paper we present a hybrid Differential Evolution (DE) algorithm for solving the frequency assignment problem (FAP) for a real GSM network. It was used a mathematical formulation adopted by other researchers [4]. In this way, it is possible the make comparisons between the results accomplished by this algorithm and others.

In this work, DE was modified to incorporate a Local Search method (i.e., optimizing the assignment of frequencies to every TRX in each sector of the solution).

The results have shown that this hybrid version of the DE algorithm, with the additional features incorporated, clearly makes possible to obtain a viable solution, when compared with other algorithms proposed by other authors. It also shows that the results of the modified DE are better compared to the original one. The proposed modifications have improved the computational efficiency of the DE algorithm to solve the approached problem.

Therefore, it is possible to conclude that the results evolution and the final results for the DE algorithm are very positives.

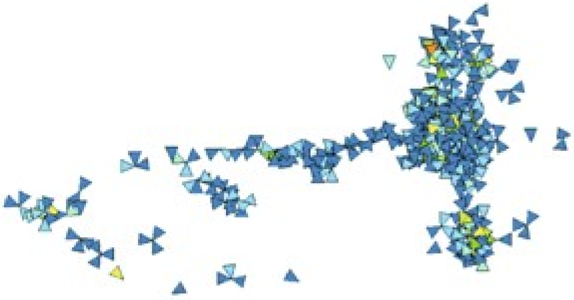


Fig 2.Topology of the GSM instance used

Future work includes the study of other evolutionary algorithms (like VNS -Variable Neighborhood Search-) to make deeper analysis with more different metaheuristics. Furthermore, we will work with more real-world instances, in order to evaluate the algorithms using different instances. Finally, the formulation of the FAP problem as a multiobjective optimization problem will be investigated as well.

## ACKNOWLEDGMENT

This work was partially funded by the Spanish Ministry of Education and Science and FEDER under contract TIN2005-08818-C04-03 (the OPLINK project). Thanks also to the Polytechnic Institute of Leiria, for the economic support offered to Marisa Maximiano to make this research.

## REFERENCES

[1] Wireless Intelligence (2006), http://www.wirelessintelligence.com

[2] C. Blum, A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”, ACM Computing Surveys 35, 2003, pp: 268–308.

[3] A. Eisenblätter, “Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds”, PhD thesis, Technische Universität Berlin, 2001.

[4] F. Luna, C. Blum, E. Alba, A.J. Nebro, “ACO vs EAs for Solving a Real-World Frequency Assignment Problem in GSM Networks.”, GECCO’07, 2007, pp: 94 – 101, London, UK.

[5] F. Luna, E. Alba, A. J. Nebro & S. Pedraza, Heidelberg, S. B. /. (ed.) Evolutionary Algorithms for Real-World Instances of the Automatic Frequency Planning Problem in GSM Networks Springer Berlin / Heidelberg, 2007, 4446/2007, 108-120.

[6] A. R. Mishra, “Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G”, chapter: Radio Network Planning and Opt., pp: 21-54. Wiley, 2004.

[7] A. M. J. Kuurne, “On GSM mobile measurement based interference matrix generation”, IEEE 55th Vehicular Technology Conference, VTC Spring 2002, 2002, pp: 1965-1969.

[8] K. Price, R. Storn., DE website. http://www.ICSI.Berkeley.edu/~storn/code.html, 2008.

[9] K. Price, R. Storn, “Differential Evolution: A Simple Evolution Strategy for Fast Optimization.”, Dr. Dobb’s Journal, 1997, 22(4): 18-24.

[10] F. Luna, C .Estébanez, et al., “Metaheuristics for Solving a Real-World Frequency Assignment Problem in GSM Networks”. GECCO’08, July 2008, pp: 1579 – 1586, Atlanta, GE, USA.

[11] M. Maximiano, M.A. Vega-Rodríguez, et al., “Analysis of Parameter Settings for Differential Evolution Algorithm to Solve a Real-World Frequency Assignment Problem in GSM Networks”, The Second International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP’08),Valencia, Spain, 2008.

[12] J. M. Chaves-González, M.A. Vega-Rodríguez, et al.. “SS vs PBIL to Solve a Real-World Frequency Assignment Problem in GSM Networks”, Applications of Evolutionary Computing. Springer. LNCS, volume 4974/2008, pp: 21-30, Berlin / Heidelberg, 2008

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 207–214

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Hierarchical Rule Design with HaDEs the HeKatE Toolchain

Grzegorz J. Nalepa
Institute of Automatics,
AGH – University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
Email: gjn@agh.edu.pl

Igor Wojnicki
Institute of Automatics,
AGH – University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
Email: wojnicki@agh.edu.pl

***Abstract*—A hierarchical approach to decision rule design in the HeKatE project is introduced in the paper. Within this approach a rule prototyping method ARD+ is presented. It offers a high-level hierarchical prototyping of decision rules. The method allows to identify all properties of the system being designed, and using the algorithm presented in this paper, it allows to automatically build rule prototypes. A practical implementation of the prototyping algorithm is discussed and a design example is also presented. Using the rule prototypes actual rules can be designed in the logical design stage. The design process itself is supported by HaDEs, a set of of design tools developed within the project.**

## I. Introduction

DESIGNING a knowledge-base for a rule-based system is a non-trivial task. The main issues regard identification of system properties, based on which rules are subsequently specified. This is an iterative process that needs a proper support from the design method being used, as well as computer tools supporting it. Unfortunately there are not many well-established tools for providing a formalized transition from vague concepts provided by the user or expert, to actual rules [1]. Quite often regular software engineering [2] methods are used [3] which are subject to so-called semantic gaps [4].

This paper focuses on transition from system properties to rule prototypes. In the paper ARD+ (Attribute Relationship Diagrams) [5], [6] a design method for decision rules is discussed. The method allows for hierarchical rule prototyping that supports the actual gradual design process. A practical algorithm providing a transition from the ARD+ design to rule design is introduced. Using ARD+ and the algorithm it is possible to build a structured rule base, starting from a general user specification. This functionality is supported by the VARDA design tool implemented in Prolog [7]. A next step is design actual rules which is aided by HQEd [8]. Both tools (VARDA, HQEd) constitute the Base of HaDEs, the *HeKatE Design Environment*.

The paper is organized in the following manner. In Sect. II the most important aspects of rule design are summarized, then in Sect. III the hierarchical approach to rule design in the *HeKatE* project is introduced. Within this approach the rule prototyping method *ARD+* is presented in Sect. V. Using the algorithm presented in Sect. VI it is possible to automatically build decision rule prototypes. A practical implementation of the prototyping algorithm is discussed in Sect. VII, with a design example in Sect. VIII. The paper ends with the concluding remarks and directions for future work in Sect. IX.

## II. Rule Design

The basic goal of rule design is to build a rule-based knowledge base from system specification. This process is a case of knowledge engineering (KE) [1], [9]. In general, the KE process is different in many aspects to the classic software engineering (SE) process. The most important difference between SE and KE is that the former tries to model how the system works, while the latter tries to capture and represent what is known about the system. The KE approach assumes that information about how the system works can be inferred automatically from what is known about the system.

In case of rules, the design stage usually consists in writing actual rules, based on knowledge provided by an expert. The rules can be expressed in a natural language, this is often the case with informal approaches such as business rules [10]. However, it is worth pointing out that using some kind of formalization as early as possible in the design process improves design quality significantly.

The next stage is the rule implementation. It is usually targeted at specific rule engine. Some examples of such engines are: *CLIPS*, *Jess*, and *JBoss Rules* (formerly *Drools*). The rule engine enforces a strict syntax on the rule language.

Another aspect of the design – in a very broad sense – is a rule encoding, in a machine readable format. In most cases it uses an XML-based representation. There are some well-established standards for rule markup languages:, e.g. *RuleML* and notably *RIF* (see www.w3.org/2005/rules).

The focus of this paper is the design stage, and the initial transition from user-provided specification that includes some *concepts*, to rule specification that connects rules with these concepts. This stage is often referred to as a *conceptual design*. It is also addressed with some recent representations, such as the *SBVR* [11] in the OMG and business rules communities.

The research presented in this paper is a part of the *HeKatE* project, that aims at providing an integrated rule design and

The paper is supported by the *HeKatE* Project funded from 2007–2009 resources for science as a research project.

implementation method for rule-based systems. The HeKatE approach to the design is briefly introduced in the next section.

## III. Hierarchical Design Approach

An effective design support for decision systems is a complex issue. It is related to design methods as well as the human-machine interface. Since most of the complex designs are created *gradually*, and they are often *refined* or *refactored*, the design method should take this process into account. Any tools aiding such a process should effectively support it.

It is worth noting that UML, the most common design method used in the SE, does not support the process at all. The process should not be mistaken with version or revision control. Simple cases of revision control are undo features, provided by most of UML tools (which is very rarely combined with real project versioning). More complex version control is usually delegated to some other, external tools in the userspace, such as CVS, Subversion, etc. But a tool supported even by a version control system is still unable to actually *present*, and *visualize* changes in the design, not mentioning its refinement or refactoring. This is related to the fact, that UML has no facilities to express the design process.

In real life the *design* support (as a process of building the design) is much more important then just providing means to visualize and construct the *design* (which is a kind of knowledge snapshot about the system). Another observation can be also made: *designing* is a knowledge-based process, where the *design* is considered a structure with procedures needed to build it (it is often the case in the SE).

In order to solve these problems, the HeKatE project aims at providing both design methods and tools that support the design process. They should be integrated, and provide a hierarchical design process that would allow for building a model, containing the subsequent design stages. Currently HeKatE supports the *conceptual design* with the ARD+ method (*Attribute Relationships Diagrams*) [5]. The main logical design is conducted with the use of XTT method (*eXtended Tabular Trees*) [12], [13].

The ARD+ design method is shortly presented in the subsequent section. It serves as a rule prototyping method for rules. The ARD+ method is a supportive design method for XTT, where the knowledge base is designed using a structured representation, based on the concept of tabular trees [14]. The XTT representation is based on the principle that the rule base is decomposed into a set of decision tables, grouping rules that have the same sets of conditional and decision attributes. It makes the knowledge base hierarchical, combining decision tables and decision trees approaches. While the use of ARD+ is aimed at XTT rules,it is a much more generic solution, allowing for prototyping different types of decision rules.

The HeKatE design process is supported by a dedicated toolchain presented in the next section.

## IV. HADES, the HeKatE Toolchain Prototype

The toolchain in the HeKatE project is composed of several elements. Currently two main design tools are available. They are integrated using XML for knowledge representation.

VARDA [7] is an ARD rule prototyping tool. It provides a basic command line interface (CLI) for design creation, refinement and visualization. The CLI is based on the Prolog interactive shell which allows calling application programming interface (API) responsible for design manipulation. It includes system modelling, input/output operations and spawning a visualization tool-chain.

The modelling is provided through several predicates performing property and attribute manipulation, as well as the transformations. The input/output operations regard storing the design as a Prolog knowledge base or exporting it into XML-based format suitable for further processing, which also serves as HQED input. It also allows to spawn the visualization toolchain based on GraphViz and ImageMagick which graphically presents the design.

VARDA is a crossplatform tool written in pure ISO Prolog, that depends only on the Prolog environment and GraphViz library. It is available under terms of the GNU GPL from https://ai.ia.agh.edu.pl/wiki/hekate:varda.

### A. HQEd

*HQeD* [8] is a CASE tool for the XTT design. The tool supports the visual design process of the XTT tables. Using the rule prototypes generated from the ARD+ design with VARDA, HQEd allows for the actual logical rule design grouped with the XTT tables.

The editor allows for gradual refinement of rules, with an online checking of attribute domains, as well as simple table properties, such as inference related dead rules. In case of simple tables it is possible to emulate and visualize the inference process step-by step. However, the main quality feature being developed is a plugin framework, that allows for integrating Prolog-based analysis plugins to check formal properties of the XTT rule base, including completeness, redundancy, or determinism.

HQEd is a crossplatform tool written in C++, that depends only on the Qt library. It is available under terms of the GNU GPL from https://ai.ia.agh.edu.pl/wiki/hekate:hqed.

The output from the editor is a complete rulebase encoded in Prolog. It can be executed using a Prolog-based inference engine. The rulebase and the inference engine can be integrated into a larger application as a logical core.

### B. XML Markup

Knowledge in the HeKatE design process is described in HML, a machine readable XML-based format. HML consists of three logical parts: attribute specification (ATTML), attribute and property relationship specification (ARDML) and rule specification (XTTML).

The attribute specification regards describing attributes present in the system. It includes attribute names and data types used to store attribute values. The attribute and property relationship specification describes what properties the system consists of and which attribute identifies these properties. Furthermore, it also stores all stages of the design process. The rule specification stores actual structured rules.

These logical parts: ATTML, ARDML and XTTML can be used in different scenarios as:

- pure ATTML – to describe just attributes and their domains,
- ATTML and ARDML combined – to describe the system being designed in terms of properties and dependencies among them,
- ATTML, ARDML and XTTML combined – attributes, dependencies and rules combined, a complete description of the system,
- ATTML and XTTML combined – just rules which are not designed out of properties, it could be used to model ad-hoc rules, or systems described by some predefined rules.

The HML, being an XML application, is formally described through a Document Type Definition (DTD). Its full specification and examples are available at https://ai.ia.agh.edu.pl/wiki/hekate:hekate_markup_language.

## V. Introduction To ARD+

The ARD+ method aims at capturing relations between *attributes* in terms of *Attributive Logic* [15]. *Attributes* denote certain system *property*. A *property* is described by one or more attributes. ARD+ captures *functional dependencies* among these *properties*. A simple property is a property described by a single *attribute*, while a complex property is described by multiple *attributes*. It is indicated that particular system property depends functionally on other properties. Such dependencies form a directed graph with nodes being properties.

A typical atomic formula (fact) takes the form $A(p) = d$, where $A$ is an attribute, $p$ is a property and $d$ is the current value of $A$ for $p$. More complex descriptions take usually the form of conjunctions of such atoms and are omnipresent in the AI literature [16], [9].

*Definition 1:* Attribute. Let there be given the following, pairwise disjoint sets of symbols: $P$ – a set of property symbols, $A$ – a set of attribute names, $D$ – a set of attribute values (the *domain*).

An attribute (see [15], [17]) $A_i$ is a function (or partial function) of the form

$$A_i\colon P_j \to D_i.$$

A generalized attribute $A_i$ is a function (or partial function) of the form $A_i\colon P_j \to 2^{D_i}$, where $2^{D_i}$ is the family of all the subsets of $D_i$.

*Definition 2:* Conceptual Attribute. A conceptual attribute *A* is an attribute describing some general, abstract aspect of the system to be specified and refined.

Conceptual attribute names are capitalized, e.g.: `WaterLevel`. Conceptual attributes are being *finalized* during the design process, into, possibly multiple, physical attributes, see Def. 8.

*Definition 3:* Physical Attribute. A physical attribute *a* is an attribute describing an aspect of the system with its domain defined.

Names of physical attributes are not capitalized, e.g. `theWaterLevelInTank1`. By finalization, a physical attribute origins from one or more (indirectly) conceptual attributes. Physical attributes cannot be finalized, they are present in the final rules.

*Definition 4:* Simple Property. $PS$ is a property described by a *single* attribute.

*Definition 5:* Complex Property. $PC$ is a property described by *multiple* attributes.

*Definition 6:* Dependency. A dependency $D$ is an ordered pair of properties $D = \langle p_1, p_2\rangle$ where $p_1$ is the independent property, and $p_2$ is the one that dependent on $p_1$.

*Definition 7:* Diagram. An ARD+ diagram $G$ is a pair $G = \langle P, D\rangle$ where $P$ is a set of *properties*, and $D$ is a set of *dependencies*.

*Constraint 1:* Diagram Restrictions. The diagram constitutes a *directed graph* with certain restrictions:

1) In the diagram cycles are allowed.
2) Between two properties only a single dependency is allowed.

Diagram transformations are one of the core concepts in the ARD+. They serve as a tool for diagram specification and development. For the transformation $T$ such as $T : D_1 \to D_2$, where $D_1$ and $D_2$ are both diagrams, the diagram $D_2$ carries more knowledge, is more specific and less abstract than the $D_1$. Transformations regard *properties*. Some transformations are required to specify additional *dependencies* or introduce new *attributes*, though. A transformed diagram $D_2$ constitutes a more detailed *diagram level*.

*Definition 8:* Finalization. The finalization $TF$ is a function of the form

$$TF : PS \to P$$

transforming a simple property $PS$ described by a conceptual attribute into a $P$, where the attribute describing $PS$ is substituted by one or more conceptual or physical attributes describing $P$. It introduces more attributes (more knowledge) regarding particular property.

An interpretation of the substitution is, that new attributes describing $P$ are more detailed and specific than attributes describing $PS$.

*Definition 9:* Split. A split is a function $TS$ of the form:

$$TS : PC \to \{P_1, P_2, \ldots P_n\}$$

where a *complex property* $PC$ is replaced by some number of *properties* ($\{P_1, P_2, \ldots P_n\}$), each of them described by one or more attributes originally describing PC. This transformation introduces more properties and defines functional relationships among them.

*Constraint 2:* Attribute Dependencies. Since $PC$ may depend on other properties $PO_1 \ldots PO_m$, dependencies between these properties and $P_1 \ldots P_n$ have to be stated.

During the design process, upon splitting and finalization, the ARD+ model grows. This growth is expressed by consecutive diagram levels, making the design more and more specific. This constitutes the *hierarchical model*. Consecutive

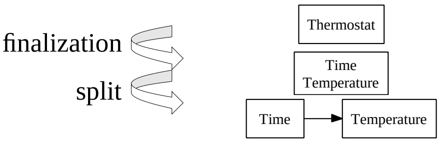


Fig. 1. Examples of finalization and split

levels make a hierarchy of more and more detailed diagrams describing the designed system. The implementation of such a hierarchical model is provided through storing the lowest available, most detailed diagram level at any time, and additional information needed to recreate all of the higher levels, the so-called *Transformation Process History* (TPH). It captures information about changes made to properties at consecutive diagram levels, carried out through the transformations: *split* or *finalization*. A TPH forms a tree-like structure then, denoting what particular property is split into or what attributes a particular property attribute is finalized into.

In Fig. 1 the first and second ARD+ design level of the thermostat example discussed in Sect. VIII are presented. These are examples of finalization and split transformations.

## VI. Rule Prototyping Algorithm

The goal of the algorithm is to automatically build prototypes for rules from the ARD+ design. The targeted rule base is structured, grouping rulesets in decision tables with explicit inference control among the latter. It is especially suitable for the XTT rule representation, however, this approach is more generic, and can be applied to any forward chaining rules.

The input of the algorithm is the most detailed ARD+ diagram, that has all of the physical attributes identified (in fact, the algorithm can also be applied to higher level diagrams, generating rules for some parts of the system being designed). The output is a set of *rule prototypes* in a very general format (`atts` stands for attributes):

```
rule: condition atts | decision atts
```

The algorithm is *reversible*, that is having a set of rules in the above format, it is possible to recreate the most detailed level of the ARD+ diagram.

In order to formulate the algorithm some basic subgraphs in the ARD+ structure are considered. These are presented in Figs. 2,3. Now, considering the ARD+ semantics (functional dependencies among properties), the corresponding rule prototypes are as follows:

- for the case in Fig. 2: `rule: e       | f, g, h`
- for the case in Fig. 3: `rule: a, b, c | d`

In a general case a subgraph in Fig. 4 is considered. Such a subgraph corresponds to the following rule prototype:

```
rule: alpha, beta, gamma, aa | bb
rule: aa                     | xx, yy, zz
```

Analyzing these cases a general prototyping algorithm has been formulated. Assuming that a dependency between two properties is formulated as: $D(IndependentProperty, DependentProperty)$, the algorithm is as follows:

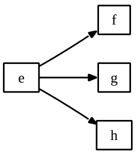


Fig. 2. A subgraph in the ARD+ structure, case #1

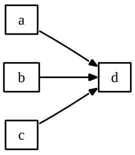


Fig. 3. A subgraph in the ARD+ structure, case #2

1) choose a dependency $D : D(F, T), F \neq T$, from all dependencies present in the design,
2) find all properties $F$, that $T$ depends on: let $F_T = \{F_{T_i} : D(F_{T_i}, T), F_{T_i} \neq F\}$,
3) find all properties which depend on $F$ and $F$ alone: let $T_F = \{T_{F_i} : D(F, T_{F_i}), T_{F_i} \neq T, /\exists T_{F_i} : (D(X, T_{F_i}), X \neq F)\}$
4) if $F_T \neq \emptyset, T_F \neq \emptyset$ then generate rule prototypes:
```
rule: F, FT1, FT2,... | T
rule: F              | TF1, TF2,...
```
5) if $F_T = \emptyset, T_F \neq \emptyset$ then generate rule prototypes:
```
rule: F | T, TF1, TF2,...
```
6) if $F_T \neq \emptyset, T_F = \emptyset$ then generate rule prototypes:
```
rule: F, FT1, FT2,... | T
```
7) if $F_T = \emptyset, T_F = \emptyset$ then generate rule prototypes:
```
rule: F | T
```
8) if there are any dependencies left goto step 1.

Rule prototypes generated by the above algorithm can be further optimized. If there are rules with the same condition attributes they can be merged. Similarly, if there are rules with the same decision attributes they can be merged as well. For instance, rules like:

```
rule: a, b | x  ;  rule: a, b | y
```

can be merged into a single rule: `rule: a, b | x, y`

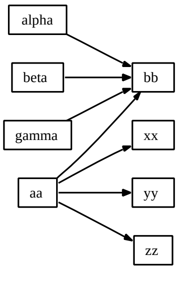


Fig. 4. A subgraph in the ARD+ structure, general case

```
ax :-
   ard_depend(F,T),
   F \= T,
   ard_xtt(F,T),
   fail.
ax.

ft(F,T,FT):- ard_depend(F,T),
   ard_depend(FT,T), FT \=F.

tf(F,T,TF):- ard_depend(F,T),
   ard_depend(F,TF), TF \= T,
   \+ ( ard_depend(X,TF), X \= F).

% generate xtt from a dependency: F,T
ard_xtt(F,T):-
   ard_depend(F,T),
   \+ tf(F,T,_),
   \+ ft(F,T,_),
   assert(xtt([F],[T])),
   ard_done([F],[T]).
ard_xtt(F,T):-
   ard_depend(F,T),
   \+ tf(F,T,_),
   findall(FT, ft(F,T,FT),ListFT),
   assert(xtt([F|ListFT],[T])),
   ard_done([F|ListFT],[T]).
ard_xtt(F,T):-
   ard_depend(F,T),
   \+ft(F,T,_),
   findall(TF, tf(F,T,TF),ListTF),
   assert(xtt([F],[T|ListTF])),
   ard_done([F],[T|ListTF]).
ard_xtt(F,T):-
   ard_depend(F,T),
   findall(TF, tf(F,T,TF),ListTF),
   findall(FT, ft(F,T,FT),ListFT),
   assert(xtt([F|ListFT],[T])),
   assert(xtt([F],ListTF)),
   ard_done([F|ListFT],[T]),
   ard_done([F],ListTF).
% retract already processed dependencies
ard_done(F,T):-
   member(FM,F),
   member(TM,T),
   retract(ard_depend(FM,TM)),
   fail.
ard_done(_,_).
```

Fig. 5. Algorithm implementation in Prolog

The rule prototyping algorithm has been successfully implemented in Prolog, as presented in the next section.

## VII. Algorithm Implementation

The ARD+ is a visual design method, so an appropriate computer tool supporting it should be developed. VARDA is a prototype ARD+ design tool written in Prolog. It is purely declarative and supports automated visualization at any design stage (diagram level) using *GraphViz* (graphiz.org). The prototype consists of an API for low-level ARD+ handling primitives such as defining attributes, properties and dependencies, and high-level primitives supporting the transformations, and visualization primitives as well. The algorithm presented here has been implemented as a part of VARDA.

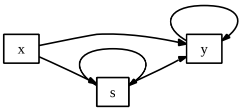


Fig. 6. Factorial, ARD+

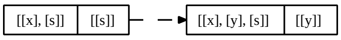


Fig. 7. Factorial, XTT prototype

The algorithm implementation (see Fig. 5) consists of: a predicate `ax/0` which spawns the search process, browsing all dependencies (implementing the first step of the algorithm), `ard_xtt/2` predicate which implements steps four through seven, and helper predicates `ft/3`, `tf/3` providing steps two and three. Additional predicate `ard_done/2` removes dependencies which have already been processed. Dependencies are given as `ard_depend/2`.

In addition to the rule prototypes some higher-level relationships can be visualized on the final design. If attributes within a set of XTTs share a common ARD+ property, such XTTs are enclosed within a frame named after the property. It indicates that particular XTTs describe fully the property. Application of this feature is subject to further research, but it seems that it is suitable for establishing rule scope, implementing the XTT *Graphical Scope* feature proposed elsewhere.

The rule generation is reversible. Having rule prototypes it is always possible to recreate the original, most detailed ARD+ level (i.e. for redesign purposes). This functionality is fully implemented in VARDA.

The algorithm has been successfully tested on several examples, including the classic *UServ Product Derby 2005* case study from the Business Rules Forum [18]. This case study includes over 70 rules.

The algorithm covers even more complicated dependencies such these in Fig. 6, including self-dependencies. This example is a model for calculating factorial iteratively. Factorial of $x$ is calculated and its value is stored in $y$, $s$ serves as a counter. The diagram could be read as follows: $y$ depends on $x$ and $s$, in addition it also depends on itself, $s$ depends on $x$ and itself.

Using the above algorithm appropriate rule prototypes is generated:

```
rule: x, y, s | y  ; rule: x, s | s
```

which can be read, that a value of $y$ is calculated based on values of $x$, $y$, and $s$. A value of $s$ is calculated based on values of $x$ and $s$.

These prototype rules are visualized in Fig. 7. Dashed arrows between XTTs indicate, in this case, that a value of $s$ is needed in order to calculate a value of $y$.

This algorithm can be applied to a design case which is presented in the next section.

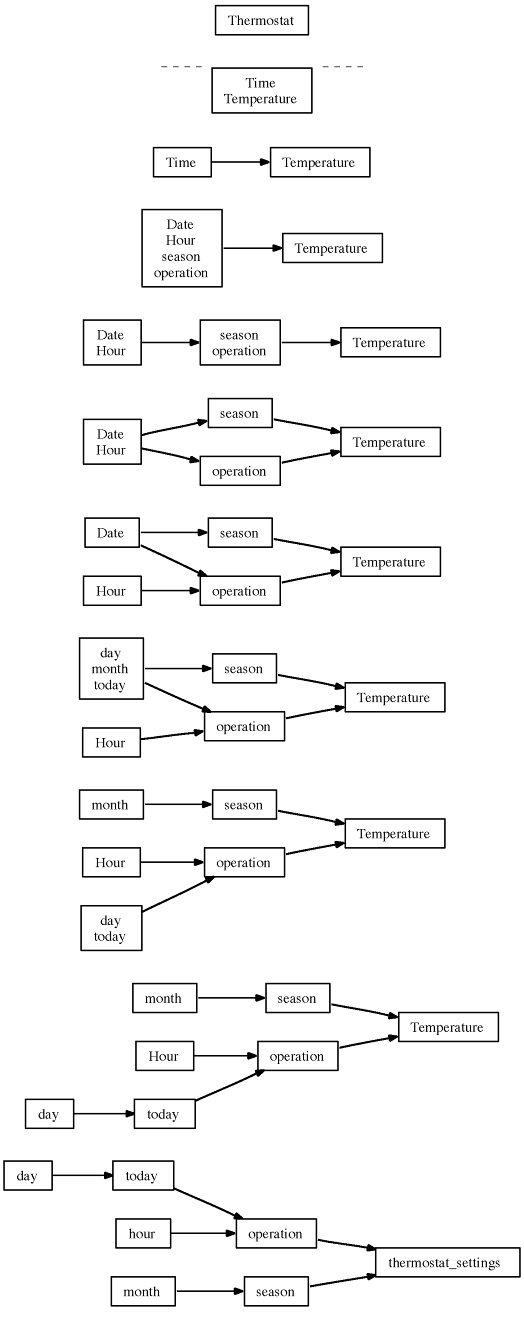


Fig. 8. Thermostat design stages, ARD+

## VIII. Design Example

A simple thermostat system is considered, see [19]. It regards a temperature control system which is to be used in an office. It controls heating and cooling which depends on several time oriented aspects. These include time of the day, business hours, day of week, month and season. Output of the system (a decision) regards what particular temperature the cooling/heating system should keep.

### A. Conceptual Design

The design process includes system property identification and refinement including property relationships, and conceptual and physical attribute identification. The design stages, consecutive ARD+ diagrams, at more and more detailed levels, are given in Fig. 8 with the most detailed diagram at the bottom. Furthermore, the example shows rule prototypes (see Sec. VIII-B) and finally the rules to control the temperature (see Sec. VIII-C). Its TPH, regarding the most detailed level, is given in Fig. 9.

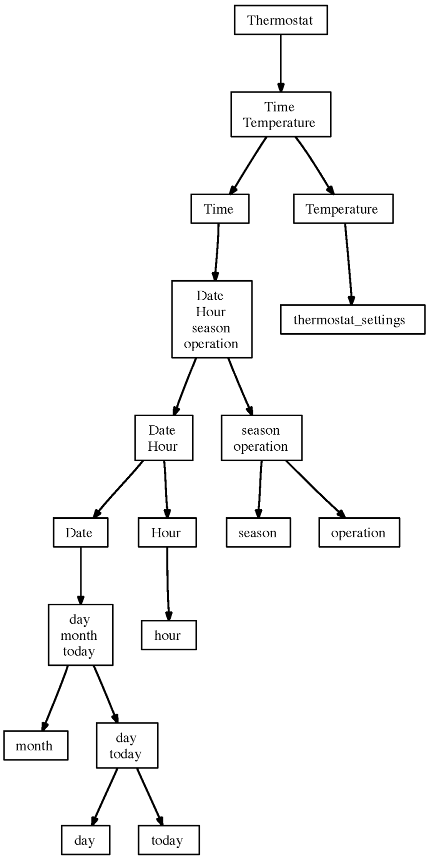


Fig. 9. The thermostat, TPH

### B. Rule Prototyping

A corresponding XTT prototype is given in Fig. 10. Higher level relationships are visualized as labeled frames, i.e.: `rule: day | today` originates from a property `Date`. Similarly, the set of XTT rule prototypes:

```
rule: day          | today
rule: month        | season
rule: today, hour | operation
```

regards `Time` property. All the prototypes regard `Thermostat`, since it is the top most system property.

Dashed arrows between XTT prototypes indicate functional relationships between them. If there is an XTT with a decision attribute and the same attribute is subsequently used by other XTT as a condition one, there is a dashed arrow between such XTTs. These relationships also represent mandatory inference control, in terms of XTT approach.

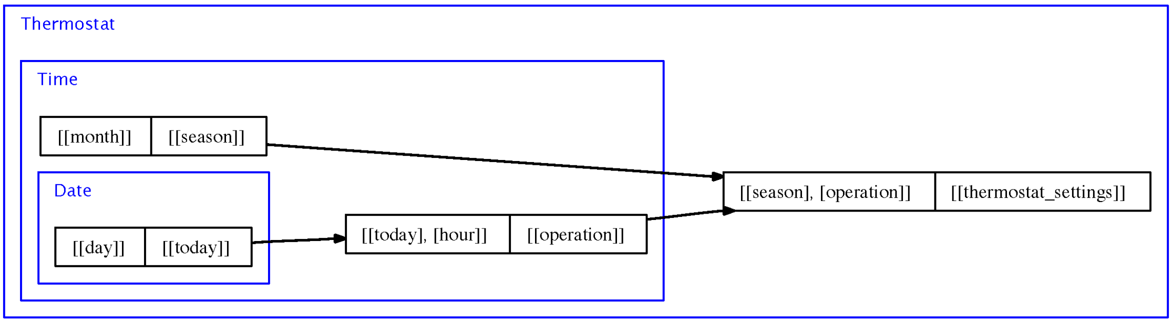


Fig. 10. The thermostat, XTT prototype

### C. Rule Design

The main logical rule design stage using HQEd is presented in Fig. 11, where the actual XTT tables corresponding to the prototypes generated using VARDA are visible.

## IX. Concluding Remarks

In the paper the hierarchical HeKatE design process is discussed. The process is supported by tools constituting the HADEs environment, these are: VARDA and HQED. VARDA is be used to assist a designer with system property identification. It can be applied as early as requirement specification, even to assist in gathering requirements. VARDA output is subsequently used as HQEd input being a template for rules. The actual rules are formulated using HQED.

The focus of the paper is on the ARD+, a hierarchical design method for rule prototyping. The original contribution of the paper is the introduction of the automatic rule prototyping algorithm based on the ARD+ design process.

Applying ARD+ as a design process allows to identify attributes of the modelled system and refine them gradually. Providing the algorithm, which builds an XTT prototype out of ARD+ design, binds these two concepts together resulting in an uniform design and implementation methodology. The methodology remains hierarchical and gradual, the algorithm works both ways: it generates rule prototypes and it is also capable of recreating the most detailed ARD+ level. Combining the most detailed ARD+ level with the TPH allows to recreate any previous ARD+ level, which provides refactoring capabilities to an already designed system.

Future work focuses on the formal definition of the inference process within XTT. There is an initial XTT inference model which requires some extensions and adaptation to cover different use scenarios. A more tight integration of the tools within HADEs is anticipated. Upon finishing it and implementing a prototype run-time environment, HADEs will provide a uniform environment for design and implementation of rule-based systems. Applications being currently considered include business applications logic [20] as well as control applications for mobile robots.

## References

[1] J. Liebowitz, Ed., *The Handbook of Applied Expert Systems*. Boca Raton: CRC Press, 1998.

[2] I. Sommerville, *Software Engineering*, 7th ed., ser. International Computer Science. Pearson Education Limited, 2004.

[3] J. C. Giarratano and G. D. Riley, *Expert Systems*. Thomson, 2005.

[4] D. Merrit, "Best practices for rule-based application development," *Microsoft Architects JOURNAL*, vol. 1, 2004.

[5] G. J. Nalepa and I. Wojnicki, "Towards formalization of ARD+ conceptual design and refinement method," in *FLAIRS-21: Proceedings of the twenty-first international Florida Artificial Intelligence Research Society conference: 15–17 may 2008, Coconut Grove, Florida, USA*, D. C. Wilson and H. C. Lane, Eds. Menlo Park, California: AAAI Press, 2008, pp. 353–358.

[6] G. J. Nalepa and A. Ligęza, "Conceptual modelling and automated implementation of rule-based systems," in *Software engineering: evolution and emerging technologies*, ser. Frontiers in Artificial Intelligence and Applications, T. S. Krzysztof Zieliński, Ed., vol. 130. Amsterdam: IOS Press, 2005, pp. 330–340.

[7] G. J. Nalepa and I. Wojnicki, "An ARD+ design and visualization toolchain prototype in prolog," in *FLAIRS-21: Proceedings of the twenty-first international Florida Artificial Intelligence Research Society conference: 15–17 may 2008, Coconut Grove, Florida, USA*, D. C. Wilson and H. C. Lane, Eds. AAAI Press, 2008, pp. 373–374.

[8] K. Kaczor, "Design and implementation of a unified rule base editor," Master's thesis, AGH Univerity of Science and Technology, june 2008, supervisor: G. J. Nalepa.

[9] A. A. Hopgood, *Intelligent Systems for Engineers and Scientists*, 2nd ed. Boca Raton London New York Washington, D.C.: CRC Press, 2001.

[10] R. G. Ross, *Principles of the Business Rule Approach*, 1st ed. Addison-Wesley Professional, 2003.

[11] OMG, "Semantics of business vocabulary and business rules (sbvr)," Object Management Group, Tech. Rep. dtc/06-03-02, 2006.

[12] G. J. Nalepa and A. Ligęza, "A graphical tabular model for rule-based logic programming and verification," *Systems Science*, vol. 31, no. 2, pp. 89–95, 2005.

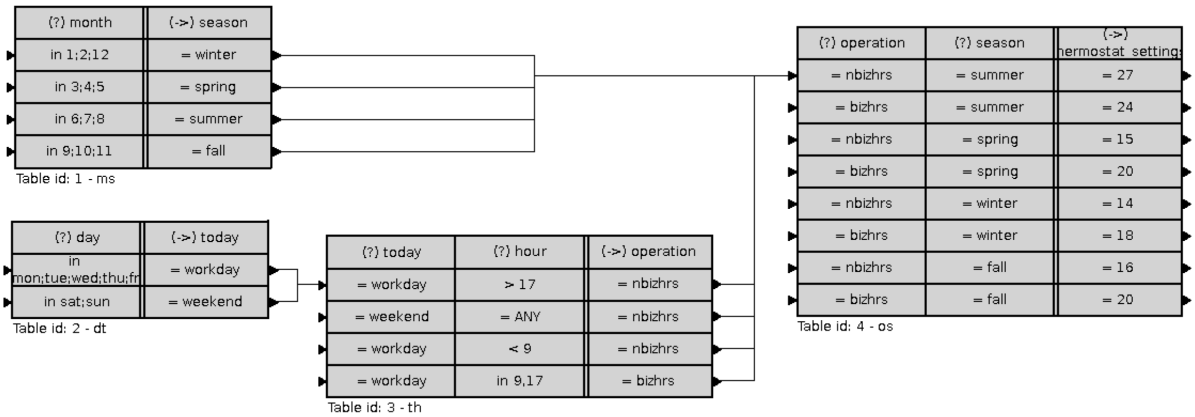
[13] G. J. Nalepa and I. Wojnicki, "Proposal of visual generalized rule programming model for Prolog," in *17th International conference on Applications of declarative programming and knowledge management (INAP 2007) and 21st Workshop on (Constraint) Logic Programming (WLP 2007): Wurzburg, Germany, October 4–6, 2007: proceedings: Technical Report 434*, D. Seipel and et al., Eds. Wurzburg: Bayerische Julius-Maximilians-Universitat. Institut fĭż¡r Informatik: Bayerische Julius-Maximilians-Universitat Wurzburg. Institut fĭż¡r Informatik, september 2007, pp. 195–204.

[14] A. Ligęza, I. Wojnicki, and G. Nalepa, "Tab-trees: a case tool for design of extended tabular systems," in *Database and Expert Systems Applications*, ser. Lecture Notes in Computer Sciences, H. M. et al., Ed. Berlin: Springer-Verlag, 2001, vol. 2113, pp. 422–431.

[15] A. Ligęza, *Logical Foundations for Rule-Based Systems*. Berlin, Heidelberg: Springer-Verlag, 2006.

[16] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall, 2003.

[17] A. Ligęza and G. J. Nalepa, "Knowledge representation with granular attributive logic for XTT-based expert systems," in *FLAIRS-20: Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference: Key West, Florida, May 7-9, 2007*, D. C. Wilson, G. C. J. Sutcliffe, and FLAIRS, Eds., Florida Artificial Intelligence Research Society. Menlo Park, California: AAAI Press, may 2007, pp. 530–535.



| (?) month | (->) season |
|---|---|
| in 1;2;12 | = winter |
| in 3;4;5 | = spring |
| in 6;7;8 | = summer |
| in 9;10;11 | = fall |

Table id: 1 - ms

| (?) day | (->) today |
|---|---|
| in mon;tue;wed;thu;fr | = workday |
| in sat;sun | = weekend |

Table id: 2 - dt

| (?) today | (?) hour | (->) operation |
|---|---|---|
| = workday | > 17 | = nbizhrs |
| = weekend | = ANY | = nbizhrs |
| = workday | < 9 | = nbizhrs |
| = workday | in 9,17 | = bizhrs |

Table id: 3 - th

| (?) operation | (?) season | (->) hermostat_setting |
|---|---|---|
| = nbizhrs | = summer | = 27 |
| = bizhrs | = summer | = 24 |
| = nbizhrs | = spring | = 15 |
| = bizhrs | = spring | = 20 |
| = nbizhrs | = winter | = 14 |
| = bizhrs | = winter | = 18 |
| = nbizhrs | = fall | = 16 |
| = bizhrs | = fall | = 20 |

Table id: 4 - os

Fig. 11. Thermostat rules

[18] BRForum, "Userv product derby case study," Business Rules Forum, Tech. Rep., 2005.

[19] M. Negnevitsky, *Artificial Intelligence. A Guide to Intelligent Systems*. Harlow, England; London; New York: Addison-Wesley, 2002, iSBN 0-201-71159-1.

[20] G. J. Nalepa, "Business rules design and refinement using the XTT approach," in *FLAIRS-20: Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference: Key West, Florida, May 7-9, 2007*, D. C. Wilson, G. C. J. Sutcliffe, and FLAIRS, Eds., Florida Artificial Intelligence Research Society. Menlo Park, California: AAAI Press, may 2007, pp. 536–541.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 215–220

ISBN 978-83-60810-14-9
ISSN 1896-7094

# On Automation of Brain CT Image Analysis

Mariusz Paradowski
and Halina Kwasnicka
and Martin Tabakov
Institute of Informatics
Wroclaw University of Technology

Jacek Filarski
and Marek Sasiadek
Department of General Radiology
Interventional Radiology and Neuroradiology
Wroclaw Medical University

***Abstract*—During recent years a number of medical diagnosis support systems have been presented. Such systems are important from medical point of view, because they aid physicians in their daily work. Some of those systems, like *Computed tomography* support systems (CT) rely on image data analysis, making it an interesting research topic from pattern recognition point of view. The paper presents a practical realization of a medical support system, including segmentation, feature extraction and decision processes. Various techniques are used within the system: fuzzy logic, shape descriptors, classifiers and automatic image annotation framework. Series of 2D CT images are an input of the system. As a result, a patient's diagnosis is presented.**

## I. Introduction

IN THE paper we present an approach to introduce automation of brain CT image analysis. Our goal is to support physicians (radiologists) in their daily work. In the first step of authors' research, the system has to classify a set of 2D CT images with respect to the presence or absence of brain atrophy. To be able to support radiologists, the constructed system has to generate diagnoses which are as similar as possible to expert's ones. To comfortably support radiologists, the system has to work within limited, reasonable time constrains. Analysis of brain CT images can be helpful in diagnosis of many diseases, including dementive diseases. Diagnosing process of dementive diseases is complicated. Although brain lesions appear 10–20 years prior to clinical manifestation, they are subtle and unavailable for standard diagnosing methods. Thus there are many researches during last few years trying to use neuroimaging procedures for diagnosing dementive diseases, such as Alzheimer's disease.

Several pattern recognition and machine vision techniques are used in the presented approach. A single 2C CT image is segmented using a fuzzy logic based algorithm.Afterward, feature vectors are calculated on the basis of extracted brain fluid segments. Classification of a patient's state is the final step. Several approaches to classification are examined. The first one is a classic multilayer perceptron (MLP). The second and the third ones are two automatic image annotation methods. Those methods are: *Binary Machine Learning* [6] using a set of *C4.5* decision trees and *Continuous Relevance Model* [4] using Gaussian kernel distance calculation. One of authors' goals during the research is to evaluate the usage of automatic image annotation framework for the stated classification problem.

The article is organized in the following manner. Next section states the problem from a medical point of view. The proposed approach is described in the third section. A brief description of all used algorithms and methods is presented. The fourth section presents performed experiments. The last section summarizes the article.

## II. Problem Statement

The most common dementia diagnosis method is visual evaluation of brain fluid spaces done by radiologist. It is quick and hardware-independent, however not perfect: it requires experienced specialist, and is descriptive and subjective [2]. More objective method is making planimetric measurements, but this is time-consuming, both during evaluation and generating results, which are relative to brain volume [5].

The most advanced method is automatic volumetric measurement of intracranial fluid. Programs used to evaluate cerebral fluid volume work on a semi-automatic basis—the specialist selects region of interest, and the specified volume is measured [7]. The next step is to make the diagnosis quicker by introduction of a fully automated method.

The automation of brain atrophy evaluation in patients with dementia on the basis of brain CT images analysis can be performed using different approaches. One is developing an effective and precise method of region of interest automatic segmentation.Afterward, region of interest volume can be measured. The second approach, proposed in this paper, is based on applied automatic annotation method which classifies brain CT images as *brain atrophy* or *no brain atrophy*. Such a method is evaluated by physicians as very helpful in dementia diagnosis process, because brain atrophy suggests dementive diseases. Taking into account above statements, the goal of this paper can be defined as follows: **To introduce a method of automatic patients discrimination with and without brain atrophy.**

The presented results are a part of wider research on CT image analysis, including image segmentation, feature generation and diagnosis support. Apart of the main goal, following research goals are defined:

- Determination if it is possible to build CT decision support system using features based on intracranial fluid area.
- Evaluation of the proposed architecture and all its components as a whole system.
- Evaluation of classification quality using features other than volume (area) of intracranial fluid.

- Evaluation of classification quality based on automatic image annotation framework in simplest 2 word case, for further research on larger dictionaries (more detailed patient's state description).

## III. Proposed approach

The goal of the proposed method is to automatically discriminate between patients with brain atrophy, which might suggest dementive disease. To achieve this a series of CT images is gathered and processed. A series of 2D CT images are used as an input of the method. Images represent brain slices, gathered during the scanning process. Together, they form a 3D CT brain image. The system output is the patient's state diagnosis.

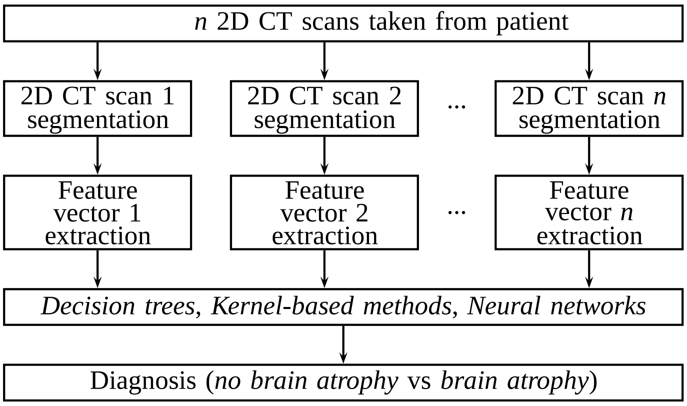

*n* 2D CT scans taken from patient
2D CT scan 1 segmentation
2D CT scan 2 segmentation
...
2D CT scan *n* segmentation
Feature vector 1 extraction
Feature vector 2 extraction
...
Feature vector *n* extraction
*Decision trees*, *Kernel-based methods*, *Neural networks*
Diagnosis (*no brain atrophy* vs *brain atrophy*)

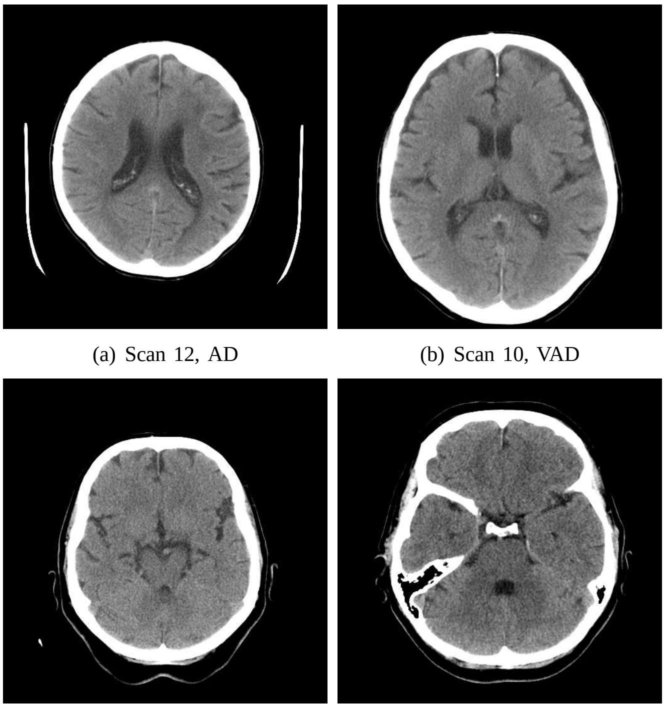
Fig. 1. Process of patients discrimination with and without brain atrophy

The proposed approach utilizes segmentation, feature vectors extraction and various classification techniques. Each of those steps is described in detail in this section. The whole process is visualized in Fig. 1. The goal of segmentation is to extract the *region of interest*, abbreviated *ROI*. Feature vectors are calculated on the basis of *ROI*. They are a set of synthesized, numerical values describing series of 2D CT scans. A set of classifiers is responsible for the final diagnosis.

### A. Image acquisition

CT scans are acquired using two-slice helical General Electric (GE) CT/e Dual scanner. Patient's head is located between X-ray source and the rows of detectors. Detectors and the lamp are rotating by 360 degrees for each scan. Acquired raw data are transformed, using Fourier transformation to 2D images. Each scan is 7 mm thick and there's no gap between the slices, so put up together all scans show complete volume of the brain and intracranial fluid. Every single pixel of 2D picture is described by value known as density, measured in Hounsfield Units (HU) and may vary between -3000 (air), through 0 (water) to 3000 (bones, metals). Only a fraction of the density range is important from the diagnostic process point of view. Original images are rescaled to the important density range and then discretized to 8-bit grayscale. As a result of image acquisition series of 512x512 8-bit grayscale images are produces.

Exemplary CT scans for four different patients are shown in Fig. 2. The first image from the left shows the 12-th scan with diagnosed *Alzheimer disease* (abbreviated *AD*). The second image shows the 10-th scan with diagnosed *Vascular Dementia* (abbreviated *VAD*). The third and the fourth scans are taken for healthy patients.


(a) Scan 12, AD
(b) Scan 10, VAD

(c) Scan 9, no disease
(d) Scan 8, no disease

Fig. 2. Exemplary CT scans for various patients. Patient (a) has *Alzheimer disease*, (b) has *Vascular Dementia*. patients (c) and (d) have no disease.

Analyzed CT scan range is also an important topic and requires discussion. As mentioned before, for each patient a series of scans is performed. CT scans taken from the bottom of the head have the lowest indexes and from the top of the head—the highest indexes. However, scans with the same indexes do not relate between patients. Differences can be quite large, mostly because of the scanning process itself. Patient's head position is one of the most problematic factors. Additionally, scans from the bottom of the head contain many artifacts, due to bone presence and scanning process. It is worth to mention, that scans do not contribute to brain atrophy analysis, equally.

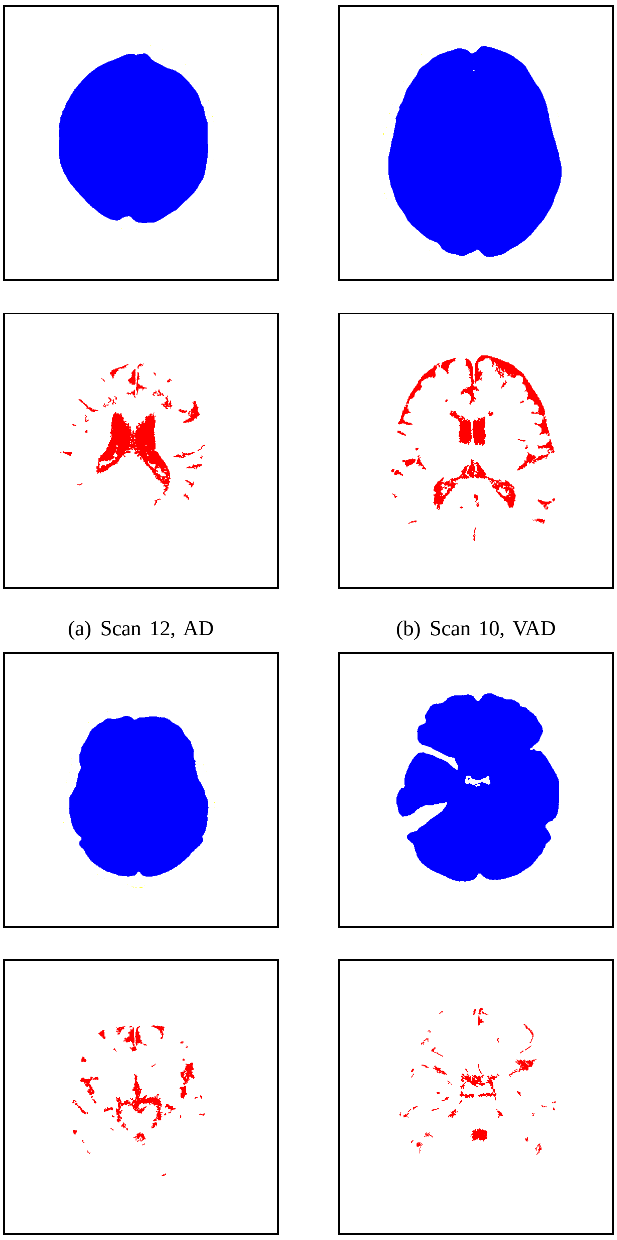
### B. Image segmentation

Acquired 2D CT scans are processed by a specialized segmentation algorithm.The segmentation process extracts *cerebrospinal fluid area*. The method emulates the complexity of the standard radiological and neurological recognition, by definition of appropriate linguistic variables in accordance with a priori introduced fuzzy rule base. In the used method the Mamdani model [1] and the Larsen implication are considered. The COA (center of area) is applied as defuzzification method.

The key point of the proposed segmentation method (considered as a region growing segmentation technique) assumes homogeneity segmentation criteria that are defined upon a fuzzy control system, dedicated to computed tomography imaging. The growing process is controlled by a fuzzy control system and thus it classifies a pixel to a certain region (defined by a *seed*) if and only if the corresponding pixel characteristics are greater than or equal to some a priori given threshold value (denoted by T), considered as an output value of the fuzzy system. A formal treatment of the growing process is given below:

Let $P$ be the set of all image pixels for a given image of size $M \times N$, $P =_{df} \{p(m,n)|m = 1,2,...,M; n = 1,2,...,N\}$ and let $P_s \subset P$ be the subset of all seeds generated for the considered image. Any neighbor pixel of a seed $p_k(x,y) \in P_s$, i.e. $p_k(x+i, y+j)$, where $i,j \in \{-1,0,1\}$ (excluding $i = j = 0$) belong to a region determined by the seed if and only if the output of the fuzzy system corresponding to this neighbor pixel is $\geq T$. In our experiments the possible primary input states for the fuzzy system (the characteristics considered for any pixel) are interpreted as the pixel colour and its localization on the image. This is because the radiological knowledge of the issue (the dementia problem) assumes that pixel represents cerebrospinal fluid if it is *dark* and it is *close* to the cranium or it is *dark* and it is *close* to the central parts of the image.

Results of the segmentation process for exemplary CT scans are presented in Fig. 3. Brain tissue is marked with blue in upper images. Cerebrospinal fluid is marked with red in lower images. The same CT scans are used as in Fig. 2.



(a) Scan 12, AD

(b) Scan 10, VAD

(c) Scan 9, no disease

(d) Scan 8, no disease

Fig. 3. Segmentation of exemplary CT scans for various patients. Upper row, blue images show the brain tissue area. Lower row, red images show the cerebrospinal fluid area.

### C. Feature vectors calculation

Feature vectors calculation is necessary for further decision process. They are one of the method's key components, because they represent the synthesized form of patient's CT image scans. A set of geometrical, real valued features is selected [3]. Feature values are calculated using extracted cerebrospinal fluid area and brain tissue area.

Two feature sets are considered. The first one, named *FS1*, contains information only about area (volume) of the cerebrospinal fluid. Absolute and relative area of the fluid is taken from each scan. This method resembles manual, volumetric approach to patient's examination (cerebrospinal fluid volume calculation). In this approach, cerebrospinal fluid areas are the basic data for the decision process.

The second one, named *FS2*, consists of many other, shape related features. Absolute and relative circumference and center of mass are taken into account. Additionally, a series of shape coefficients are used. All used features are presented in Tab. I.

Patient's examination results in a series of 2D CT scans. On the basis of acquired and segmented CT scans a set of feature vectors is calculated. Each feature value is calculated for the whole CT image using the segmentation output. The process results in a non-square matrix of values generated for one patient. The following symbols are used in the given definitions:

– $i$—number of pixel within object,
– $L$—circumference of cerebrospinal fluid,
– $S$—area of cerebrospinal fluid,
– $L_B$—circumference of brain tissue,
– $S_B$—area of brain tissue,
– $r_i$—distance of a pixel from the center of the mass,
– $d_i$—distance of a contour pixel from the center of the mass,
– $l_i$—minimum pixel distance to the object contour,

TABLE I
FEATURES CALCULATED FOR A SINGLE, SEGMENTED CT SCAN. *FS1* REPRESENTS THE FIRST USED FEATURE SET, *FS2* REPRESENTS THE SECOND ONE.

| *Feature* | *Definition* | *FS1* | *FS2* |
|---|---|---|---|
| Number of scan | $n$ | yes | yes |
| Absolute circumference | $L$ | no | yes |
| Absolute area | $S$ | yes | yes |
| Relative circumference | $L_{rel} = \frac{L}{L_B}$ | no | yes |
| Relative area | $S_{rel} = \frac{S}{S_B}$ | yes | yes |
| Malinowska Coefficient | $R_M = \frac{L}{2\sqrt{\pi S}} - 1$ | no | yes |
| Modified Malinowska Coeff. | $R_{mM} = \frac{2\sqrt{\pi S}}{L}$ | no | yes |
| Blair-Bliss Coefficient | $R_B = \frac{S}{\sqrt{2\pi(\sum_i r_i)^2}}$ | no | yes |
| Blair-Danielsson Coeff. | $R_D = \frac{S^3}{\left(\sum_i l_i\right)^2}$ | no | yes |
| Haralick Coefficient | $R_H = \sqrt{\frac{(\sum_i d_i)^2}{n \sum_i d_i^2 - 1}}$ | no | yes |
| Feret Coefficient | $R_F = \frac{L_h}{L_v}$ | no | yes |
| Circularity Coefficient 1 | $R_{C1} = 2\sqrt{\frac{S}{\pi}}$ | no | yes |
| Circularity Coefficient 2 | $R_{C2} = \frac{L}{\pi}$ | no | yes |
| Center of mass in X | $R_{mx} = \frac{\sum_i x_i}{\sum_i 1}$ | no | yes |
| Center of mass in Y | $R_{my} = \frac{\sum_i y_i}{\sum_i 1}$ | no | yes |

– $L_h$—maximum width of the object,
– $L_v$—maximum height of the object.

$L$ and $L_B$ values are calculated using the following approach. Each pixel in the image, excluding image boundary pixels, is analyzed if it is a ROI boundary pixel. Only ROI boundary pixels are considered in the circumference value calculation. The process is a lookup procedure using a $3 \times 3$ pixels mask. A pixel is ROI boundary if and only if the center pixel is lit and at least two pixels in the masked area are not lit. In case not lit masked pixels are placed on the diagonal, the pixel contributes $\sqrt{2}$ to the circumference value, in all other cases it contributes $1$.

Normalized feature vectors are required by classification methods. All presented features are normalized using linear transformation to $\langle 0; 1\rangle$ interval. Each feature's lower and upper normalization bound is selected as the lowest and highest value from the available data.

### D. Decision making

The last step performed by the proposed medical support system is decision making. Three approaches are presented and evaluated. The first one is a classic multilayer perceptron neural network. The second and the third methods are based on automatic image annotation framework. All used methods are supervised machine learning and require a training set. Training examples are a set of manually diagnosed CT images. Manual diagnoses are performed by the radiologist and thus they represent medical domain knowledge.

*1) Classification using multilayer perceptron:* The first approach to decision making is a classic multilayer perceptron. In the presented approach, 3-layer perceptron is used. Number of neurons in the input layer is equal to the number of features in a single feature vector. Number of neurons in the output layer is equal to the number of output classes. Size of the hidden layer is calculated using a standard rule and is equal to the average size of input and output layer. Number of training epochs is equal to 500, learning rate is equal to 0.3 and momentum coefficient to 0.2.

Final decision process is the following. Each feature vector is classified using the neural network. Results are aggregated and the dominant class in all classification runs is selected.

*2) Classification using automatic image annotation approach:* In general, an automatic image annotation method describes an input image (represented as a set of feature vectors) with a subset of words from the given *dictionary*. Input of image annotation is a set of feature vectors and a dictionary. Output of automatic image annotation is a subset of dictionary. The goal of the method is to select the subset in best possible way, according to presented *training examples*. In this case, the dictionary is reduced to only two possible choices: *brain atrophy* and *no brain atrophy* (*brain atrophy* is treated as a single word, similarly as no brain atrophy). Output of the method is also reduced to only one word, being the diagnosis. Our goal is to verify if automatic image annotation framework can be used for the stated classification problem. Successful verification will lead to further research, with larger dictionaries.

Two automatic image annotation methods are examined for the stated problem. The first one is *Continuous Relevance Model* [4], abbreviated *CRM*. It is based on distance calculation between processed feature vector and feature vectors from the training set. This method uses distance measure based on *Gaussian kernel*. Words are propagated from the training set into the processed feature vectors with respect to the measured distance. The farther the feature vectors are, the lesser the influence is.

The second method is *Binary Machine Learning* [6], abbreviated *BML*. It employs a series of decision trees, constructed using *C4.5* algorithm. Each decision tree relates to one class from the dictionary and is responsible for generation of positive or negative answers. A positive answer given by a decision tree contributes to the word the tree represents. If the tree gives a negative answer, no word contribution is made. In case of $n$ words (in this case $n = 2$), $n$ decision trees is required.

## IV. EXPERIMENTS

The goal of conducted experiments is to show the method's ability to automatically discriminate between *brain atrophy* and *no brain atrophy* CT images. Experiments are performed on a set of CT images acquired from patients with various types of brain disorder and with no disorder. Segmentation and feature calculation are performed according to the presented routines. Three methods of patients discrimination are tested: *Multilayer perceptron neural network*, *Continuous Relevance Model* and *Binary Machine Learning*.

### A. Medical data and quality evaluation

Available medical data are divided into four categories according to the disease type (Tab. II). In each category a number of patients are examined. There are total 72 examinations available. Patients with three disease types are included in the performed experiments: *Alzheimer disease, Mixed Dementia* and *Vascular Dementia*. Patients with no brain atrophy are the fourth group. Examined patients are in different age and have different education level.

TABLE II
NUMBER OF PATIENTS ACCORDING TO THE DISEASE TYPE. CLASSIFICATION USED IN THE AUTOMATED PROCESS IS PRESENTED IN THE THIRD COLUMN.

| *Disease type* | *Number of patients* | *Class* |
|---|---|---|
| No disease | 20 | no brain atrophy |
| Alzheimer disease | 35 | brain atrophy |
| Mixed Dementia | 12 | brain atrophy |
| Vascular Dementia | 5 | brain atrophy |

Results quality evaluation is done with respect to manually prepared diagnoses. Those diagnoses are the reference point both in training and testing processes. All experiments are performed using *leave-one-out* validation procedure. The following values are calculated to evaluate quality: true positives, false positives, true negatives, false negatives, false positives ratio, false negatives ratio and overall accuracy.

To make this discussion more complete, we also address the accuracy baseline. Baseline value for accuracy is equal to 72% and it is calculated for a naive approach. To maximize accuracy, every case is classified to the most common class, without any feature consideration. For the presented data, the most common class is *brain atrophy*. This means that there are only positive answers (true positives and false positives) and no negative answers.

### B. Results

Manual diagnosis is performed by medical doctors, radiologists specialized in CT image analysis. This approach is the reference one. It requires a lot of manual labor, thus is expensive and prone to potential errors. Automated diagnosis is performed using the proposed method. Detailed information on classification results are presented in Tab. III. True positives, true negatives, false positives and false negatives are listed. Selection of feature set (*FS1* or *FS2*) has the largest influence on the achieved results. Selection of scan range and classification method have smaller influence. According to the performed experiments, usage of *FS2* results in much higher quality of results than usage of *FS1*. *BML+FS1* gives a large number of type II errors (false negatives). *CRM+FS1* gives a large number of type I errors (false positives). In one case, true negatives and false negatives have not been generated at all, which means that the method have not generated any negative answer. *MLP+FS1* approach fails on generation of negative answers and proves to be unusable.

As mentioned before, usage of *FS2* results in much better results. Number of true positives for both combinations (*CRM+FS2* and *BML+FS2*) is comparable or larger than for *CRM+FS1*, and number of true negatives is comparable to *BML+FS1*.

TABLE III
BRAIN ATROPHY VERSUS NO BRAIN ATROPHY DETAILED CLASSIFICATION RESULTS FOR EXAMINED APPROACHES. TP STANDS FOR TRUE POSITIVES, TN—TRUE NEGATIVES, FP—FALSE POSITIVES, FN—FALSE NEGATIVES.

| *Used approach* | *TP (correct)* | *FP (type I error)* | *TN (correct)* | *FN (type II error)* |
|---|---|---|---|---|
| Scan range: All | | | | |
| CRM+FS1 | 30 | 10 | 10 | 22 |
| BML+FS1 | 35 | 2 | 18 | 17 |
| MLP+FS1 | 51 | 16 | 4 | 1 |
| CRM+FS2 | 49 | 4 | 16 | 3 |
| BML+FS2 | 51 | 6 | 14 | 1 |
| MLP+FS2 | 48 | 3 | 17 | 4 |
| Scan range: 5-15 | | | | |
| CRM+FS1 | 52 | 20 | 0 | 0 |
| BML+FS1 | 33 | 2 | 18 | 19 |
| MLP+FS1 | 52 | 20 | 0 | 0 |
| CRM+FS2 | 50 | 5 | 15 | 2 |
| BML+FS2 | 52 | 13 | 7 | 0 |
| MLP+FS2 | 49 | 7 | 13 | 3 |
| Scan range: 10-15 | | | | |
| CRM+FS1 | 42 | 13 | 7 | 10 |
| BML+FS1 | 32 | 4 | 16 | 20 |
| MLP+FS1 | 52 | 20 | 0 | 0 |
| CRM+FS2 | 46 | 5 | 15 | 6 |
| BML+FS2 | 50 | 6 | 14 | 2 |
| MLP+FS2 | 49 | 8 | 12 | 3 |
| Scan range: 12-15 | | | | |
| CRM+FS1 | 50 | 20 | 0 | 2 |
| BML+FS1 | 35 | 4 | 16 | 20 |
| MLP+FS1 | 52 | 20 | 0 | 0 |
| CRM+FS2 | 47 | 2 | 18 | 5 |
| BML+FS2 | 49 | 7 | 13 | 4 |
| MLP+FS2 | 50 | 4 | 16 | 2 |

Achieved accuracies and negative rates are presented in Tab. IV. Averaged results are the confirmation of the discussion, presented above. Usage of *FS2* in general leads to better results, comparing to *FS1*. In terms of accuracy, selection of the annotation method do not have large influence on the quality. Presented results suggest that *MLP+FS2* with scan range *12-15* should be selected as the best approach. Usage of *MLP+FS2* with scan range *12-15* results in accuracy equal to 92%. *CRM+FS2* approach also provided good results. Accuracy is equal to 90% (only 1 more patient is incorrectly classified, compared to *MLP+FS2*), false positives and false neatives rates are both equal to 10%.

### C. Discussion

Achieved results are satisfying and can be viewed as a good basis for further research. When provided with a proper feature set, all three approaches to classification give similar results. There is a large difference in results quality between tested feature sets. The first one, which is based only on cerebrospinal fluid areas, provides unsatisfactory results. However, when the feature vector is extended, quality is increased. Examined scan ranges do not influence the results quality in a large manner. Best results are achieved for scan range 12-15, however usage of all scans gives only very minor decrease in quality (1 more patient is misclassified).

TABLE IV
BRAIN ATROPHY VERSUS NO BRAIN ATROPHY AVERAGED CLASSIFICATION RESULTS FOR EXAMINED APPROACHES.

| *Used approach* | *FP rate* | *FN rate* | *Accuracy* | *Relation to baseline* |
|---|---|---|---|---|
| Scan range: All | | | | |
| CRM+FS1 | 0.50 | 0.42 | 0.56 | below baseline |
| BML+FS1 | 0.10 | 0.33 | 0.74 | above baseline |
| MLP+FS1 | 0.80 | 0.02 | 0.76 | above baseline |
| CRM+FS2 | 0.20 | 0.06 | 0.90 | above baseline |
| BML+FS2 | 0.30 | 0.02 | 0.90 | above baseline |
| MLP+FS2 | 0.15 | 0.08 | 0.90 | above baseline |
| Scan range: 5-15 | | | | |
| CRM+FS1 | 1.00 | 0.00 | 0.72 | baseline case |
| BML+FS1 | 0.10 | 0.37 | 0.71 | below baseline |
| MLP+FS1 | 1.00 | 0.00 | 0.72 | baseline case |
| CRM+FS2 | 0.25 | 0.04 | 0.90 | above baseline |
| BML+FS2 | 0.65 | 0.00 | 0.82 | above baseline |
| MLP+FS2 | 0.35 | 0.06 | 0.86 | above baseline |
| Scan range: 10-15 | | | | |
| CRM+FS1 | 0.65 | 0.19 | 0.68 | below baseline |
| BML+FS1 | 0.20 | 0.38 | 0.67 | below baseline |
| MLP+FS1 | 1.00 | 0.00 | 0.72 | baseline case |
| CRM+FS2 | 0.25 | 0.11 | 0.85 | above baseline |
| BML+FS2 | 0.30 | 0.04 | 0.89 | above baseline |
| MLP+FS2 | 0.40 | 0.05 | 0.85 | above baseline |
| Scan range: 12-15 | | | | |
| CRM+FS1 | 1.00 | 0.04 | 0.69 | below baseline |
| BML+FS1 | 0.10 | 0.33 | 0.74 | above baseline |
| MLP+FS1 | 1.00 | 0.00 | 0.72 | baseline case |
| CRM+FS2 | 0.10 | 0.10 | 0.90 | above baseline |
| BML+FS2 | 0.35 | 0.06 | 0.86 | above baseline |
| MLP+FS2 | 0.20 | 0.04 | 0.92 | above baseline |

Let us relate to the stated research goals, now. Construction of a CT decision support system is possible, however great care has to be made during feature selection process. Areal features seem to be insufficient and other shape related features must be introduced. The presented support system architecture gives satisfactory results. Having the accuracy baseline value equal to 72%, authors have achieved maximum classification accuracy reaching 92%. Verification of automatic image annotation framework is also positive. Achieved results are similar to those with standard classification approach.

## V. SUMMARY

A method for automatic discrimination of brain-atrophic patients with possible dementive disease and patients with no brain atrophy is proposed. A series of 2D CT scans, representing the 3D brain image is given as input. Each scan is segmented by a specialized routine, extracting the cerebrospinal fluid and nervous tissue. A set of feature vectors is calculated on the basis of segmentation and they are input of a classification method. Afterward, feature vectors are processed and a diagnosis is generated.

Achieved results are very promising. For the best presented combination of feature set, scan range and classification method, accuracy reaches 92%. It means that the automatic method is able to repeat expert's manual diagnoses in most cases. Additionally, the automated process is not time consuming and can be used as a live aid in medical doctors daily work.

The proposed method could be also used for prospective and populational researches due to its speed and repeatability of output scores. Another possible usage is the evaluation of possible changes in atrophy degree, e.g. during treatment—especially in experimental methods of treatment. Further research will be focused on introduction of new, shape related features characterizing the cerebrospinal fluid. Additionally, a larger dictionary (with more detailed diagnoses) will be proposed for the image annotation approach.

## ACKNOWLEDGMENT

This work is financed from the Ministry of Science and Higher Education resources in 2006 – 2008 years as a research project N518 022 31/1338.

## REFERENCES

[1] Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies,* Vol. 7, No. 1, pp. 113, 1975.

[2] Leonardi M, Ferro S, Agati R.: Interobserver variability in CT assesment of brain atrophy. *Neuroradiology,* No. 36, pp. 17-19, 1994.

[3] Tadeusiewicz R.: Komputerowa analiza i przetwarzanie obrazow. Wydawnictwo Fundacji Postepu Telekomunikacji, Krakow, 1997 (in Polish).

[4] Lavrenko V., Manmatha R., Jeon J.: A Model for Learning the Semantics of Pictures, In Proc. of NIPS'03, 2003.

[5] Frisoni GB, Scheltens P, Galuzii S, Nobili FM, Fox NC, Robert PH, Soininen H.: Neuroimaging tools to rate regional atrophy, subcortical cerebrovascular disease and regional cerebral blood flow and metabolism: consensus paper of the EADC. *J Neurol. Neurosurg. Psychiatry* No. 74: pp. 1371-1381, 2003.

[6] Kwasnicka H., Paradowski M.: Multiple Class Machine Learning Approach for Image Auto-Annotation Problem, ISDA 2006 *Proceedings, IEEE Computer Society,* Vol. II, pp. 353-361, 2006.

[7] Czarnecka A., Sasiadek M.J., Hudyma E., Kwasnicka H., Paradowski M.: Computer-Interactive Methods of Brain Cortical and Subcortical Atrophy Evaluation Based on CT Images, 2008, Springer-Verlag (in printing).

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Neuronal Groups and Interrelations

Michał Pryczek
Institute of Computer Science
Technical University of Lodz
Wolczanska 215, 90-924 Lodz, Poland
Email: michalp@ics.p.lodz.pl

***Abstract*—Recent development of various domains of *Artificial Intelligence* including *Information Retrieval* and *Text/Image Understanding* created demand on new, sophisticated, contextual methods for data analysis. This article formulates *Neuronal Group* and *Extended Neuron Somatic* concepts that can be vastly used in creating such methods. Neural interrelations are described using graphs, construction of which is done in parallel with neural network learning. Prototype technique based on *Growing Neural Gas* is also presented to give more detailed view.**

***Index Terms*—neural networks, pattern recognition, structural pattern recognition, pattern classification.**

## I. Introduction

IN GENERAL, modern research on classifiers tends to develop into three non excluding directions. The first one, commonly known as *Kernel Methods* performs kernel based indirect mapping to possibly unknown feature space, known as *kernel trick*. The most notable representatives of this group are *Support Vector Machines* [1]. The second direction makes use of *classifier committee* concept, giving *AdaBoost* and others [2], [3] as well recognized examples.

These techniques can with some success be applied also to structural pattern recognition task next to dedicated solutions. Commonly considered tasks of this family are text and image segmentation. The one plays vital role in *Information Retrieval* field, the other in dynamically developing *Image Understanding* [4] paradigm. New tasks in these fields, like *Opinion Mining* [5] and *Image Symbolisation* [4] prove the amount of work that is still to be done, yielding new models, concepts and techniques that would face problem complexity.

Modern probabilistic models of structural pattern recognition, namely family of methods referred to as *Conditional Random Fields* [6] prove vitality of contextual approach to these problems [7]. Advances made in this field in recent years encourage exploration of possible expert knowledge and context utilisation. What is important, this is done outside *vector space* model of object representation, often referred to as *feature space*, that is the most widely used in *Artificial/Computational Intelligence* community.

Section II contains general discussion about data handling different kinds of knowledge utilization while computing object representation. Section III describes conceptual model of *neuronal group* and *graph based description* of *neuronal interrelations*. This kind of knowledge about data is inferred during neural network learning, building a context that can be used during the rest of learning phase as well as can form part of the solution to the addressed problem.

A prototype method taking benefits from *Neuronal Group Learning* concept is defined in section IV using *Adaptive Active Hypercontour* [8] frame. Other possibilities and promising learning process duality considerations are presented in section V. Supervised classification task will be referred to as an example field of applications.

## II. Extended Neuron Somatic Description

Analysing most of contemporary *pattern analysis* methods one can come to the conclusion, that in almost all cases input space definition plays crucial role in entire process.

### A. Object space transformation

Construction of *input space* $\mathcal{X}$ can be usually described as object transformation from domain space $\mathcal{O}$ to $\mathcal{X}$, here denoted as a function $\phi : \mathcal{O} \rightarrow \mathcal{X}$ [9]. It often includes sophisticated, domain dependent feature extraction and selection techniques. It is typically required, that $\mathcal{X}$ fulfils all requirements of the algorithm we wish to use to analyse it. This usually means, that $\mathcal{X}$ is either *inner product*, *normed* or *metric* space. Various types of elements' *similarity* can also be found in literature and are enough for some methods.

Meaningful construction of $\mathcal{X}$ together with target operations on space elements (e.g. metric) are hard tasks, requiring substantial domain knowledge. Their vitality in entire process can additionally be emphasised by the fact, that mistakes done on this phase of research usually cannot be overcome by choosing more powerful algorithms which operate on $\mathcal{X}$, as they cannot supplement the knowledge that has been lost during transformation $\phi$ (if it was not redundant).

One of the most popular approach to input space definition is so called *vector space model* (VSM) - considering $\mathcal{X} = \mathbb{R}^n$ for some natural $n$. Its usefulness cannot be underestimated, especially that its properties have been subjected to many research initiatives and successfully applied to many tasks. The most important benefit of *VSM* usage is availability of numerous standard operation definitions (including sophisticated inner products and metrics) that can be directly used on input space elements regardless $\phi$ actual definition. On the other hand, this model puts extreme importance on feature extraction, selection and informative power of feature set. The other approach considers more complex representation domains, e.g. including graph structures. It aims to preserve more information stored in original object representation in set $\mathcal{O}$. However, more sophisticated and tightened to $\mathcal{X}$ operations are required. In the end, this approach needs similar infield expert knowledge level, which is introduced in $\mathcal{X}$ operation level rather than in $\phi$ transformation definition. Last years of research on *structural kernels* replacing inner products brought huge progress in this field.

Even though vital to final results, *input space* definition utilises mostly expert/engineer knowledge and experience. It is worth mentioning, that $\phi$ usually consists of subsequent steps. Basic feature extraction can, and usually is followed by operations like weighting, normalization, feature selection or chained feature extraction. This transformations often utilise additional knowledge coming not only from expert, but also from data. It is usually inferred from some set of objects accompanied by additional information, e.g. considering supervised classification task this role is typically played by training set, or elements of specially prepared tuning set. It is surprising, that extraction and utilisation of knowledge about interrelations of different areas of $\mathcal{X}$ did not find equally big interest among research teams. One of the efforts worth mentioning are contextual textual document processing ideas presented in [10]

### *B. Extensions In Neuron Somatic Description*

Let the input space be real vector space, to comply with classic *Hebb*'s model of neuron [11]. Somatic parameters of neuron are also required to be real numbers, with additional parameter—*bias weight*. However, other phenomena can have impact on neuron excitement (see *lateral feedback* example in section III). The biggest potential lies in sensor neurons, which operate directly on input data. Currently, somatic information (parameters) stored by neuron are the one required to compute its activity after input presentation. Reminding *Hebb*'s model these are weights (one value for each input including *bias*). There are also other factors that directly or not can influence neuron's behaviour, like position in neural cortex (absolute or relative to neighbouring neurons), activation function parameters not related to input etc. Neuron potential used in *Kohonen* networks to deal with dead neuron problem shows commonness of this approach, which capabilities seems to be underestimated by the researchers.

Only some of these additional parameters can properly be considered *somatic* in biological meaning of this word. Nevertheless, it is proposed to extend somatic description of neuron by data that not necessarily, as input weights or even mentioned *potential*, have direct influence on excitement computation. In contrast, they may play important role during network model adaptation or final result formulation, e.g. by defining context of computation. See sections III and V for more details.

## III. Neuronal Groups

Let neural network be seen as a set of neurons $V$, their parameters and some kind of its architecture description $A$, that defines *synaptic* neuronal interconnections. These connections meet reflection in somatic description of adjacent neurons. In the concept presented it is proposed, that every set $G \subseteq V$ can form a *neuronal group*. Internal relations between group members and with other groups can be used in both network output computation and its learning.

### *A. Neurophysiological Background: Neuronal Group Selection Theory (NGST)*

Rapid development of neurophysiology in $20^{th}$ century brought numerous theories about how brain is organised and cognitive process proceed. One of the most interesting and revolutionary ones is called *Neural Darwinism* [12]. Even though concepts presented do not refer directly to evolutionary process proposed by *Edelman*, it may be considered a neurophysiological and, more notably, *philosophical* analogy to the presented work.

The *Neuronal Group Selection Theory* states [12], that brain is dynamically organised into (sub)networks, structure and function of which arises from its development and behaviour. These networks, consisting of thousands of strongly interconnected neurons, are called *neuronal groups* and considered as *functional units*. *Edelman* considers three phases/processes during neural map development. The first one, aimed to build *primary repertoire* of *neuronal groups*, is done within neurobiological ant anatomical constraints dictated by genotype. *Primary variability* of neurons and their synaptic connections form a basis to further process of self-organisation. This phase takes place mostly during foetal life and infancy, when neural system is mainly exposed to self-generated activity and consequently self-afferent information. Subsequent experiment based selection catalyses development of best performing *group prototypes*, leading to formation of primary *functional units* (*primary neural repertoire*). Even though genetically and anatomically determined, adaptation and selection process is itself epigenetic and it is extremely improbable for two organisms to have identical neural structure- this is called *primary variability*.

Development of neural system unavoidably leads to increase of quantity and variability in information processed by it. During postnatal life an epigenetic process of synaptic weight adaptation takes place, leading to composition of *primary groups* into more complex structures resulting in forming *secondary neural repertoire* in a subsequent selective process. This phase is even more ontogenetic, as it bases on individual experiences which create different predicates for *neuronal group selection*. Temporal correlations between various groups leads to creation of dynamically emerging neural maps, whose re-entrant interconnections maintain spatio-temporal continuity in response to re-entrant signals. A resulting neural maps can be considered an effective functional units of brain.

Seeking for a more detailed and intuitive example, please refer to *Mijna Hadders-Algra*'s article [13], which presents human motor development in light of *NGST*.

The idea presented in this article is convergent with *Neuronal Group* concept presented by *Edelman* and thus shares its name. However, at least two important differences must be outlined. The first one is that in the concept presented *neuronal groups* do not have to reflect degree of interconnection between neurons from the group; various approach to group distinguishing was presented in subsection III-C. The second difference is related to network and group development. Evolutionary process analogous to *NGST* might be considered as one of numerous possible algorithms and does not exhaust concept perspectives.

### B. Lateral Feedback

An other inspiration for presented model was *lateral feedback* mechanism and facts about formation of cortical map [11]. *Lateral feedback* refers to observed impact which activity of latent neurons have on other neuron excitement and plays important role in topographic organisation of cerebral cortex. Depending on selected interaction model we generally can have positive and negative influence (increasing and decreasing neuron activation), which usually reflects distance between neurons. Dynamical properties lead to formation of bubbles-groups of neurons that tend to recognise similar signals.

This very simple mechanism, although impractical from computational point of view because of dynamical stability problems, focuses on cooperation of neurons and group (bubble) activity as potential research field. This biological motivation can be presented as inspiration of widely used self-organising neural networks called *Self Organising Feature Maps*, *Kohonen networks* [11] and similar derivative models. However, the approach presented addresses different aspects of neural interactions, possibly receding physiological inspirations.

### C. Neuronal Group concept

Let $\mathbb{S}$ denotes a set of *neuronal groups*, which can also be considered as a set of group labels. In the concept presented all neurons $n \in V$ have associated set of groups to which they belong, here denoted as function $l : V \to 2^{\mathbb{S}}$, also referred to as *labelling of neurons*. It can be considered *somatic description extension* (see section II-B) but this is not crucial to the expression of the idea. Limitations on $l$'s codomain can be imposed to simplify model or tune it to specific needs. For instance, condition

$$\forall_{n\in V}\overline{\overline{l(n)}} = 1 \qquad (1)$$

enforces mutual exclusions of groups and can also be described functionally as $\hat{l} : V \to \mathbb{S}$. In practice, it is desired to introduce or infer groups of different kinds, playing different roles in algorithms that can utilise information coming from grouping of neurons. As it will be shown later, mutual exclusion condition is important tool, but to avoid tainting the concept it should be imposed only on subsets of $\mathbb{S}$, in cases when these group distinguishing policies exclude themselves.

In general we can consider three possible approaches to group identification:

1) predefined groups—identified by an expert/engineer contexts in which interrelations of neurons should be considered.
2) supervised group identification—in which an external premise/knowledge to distinguish groups or perform neuron grouping; this is typical situation in supervised classification tasks, where category labels can act as group descriptors.
3) unsupervised group identification—possibly the most interesting approach; similarly to object grouping tasks, the number of groups can be predefined or not.

*Predefined* groups might be intentionally compared to *primary neuronal repertoire* included in *NGST*, whereas the other two corresponds to *secondary repertoire*. *Unsupervised group identification* is in general correspondence with *NGST* in approach and philosophy, as premises to groups selection are formed by *internal* relations between neurons and inputs (experiences) related phenomena. However, logical weakness of such analogies should be emphasised.

### D. Neuronal interrelations and their descriptions

The most natural way to describe relations between neurons is using graphs of different kind. Obviously, entire network architecture $A$ can be described as weighted, directed graph. Another example of the often analysed relation type would be topological neighbourhood, utilised by self-organising neural networks during learning process, naming *Growing Neural Gas* algorithm (*GNG*) as most straightforward implementation of this idea.

Going further, let us consider set $\mathcal{G}$ as set of available graph interrelation descriptions. Graphs $G \in \mathcal{G}$ can differ on type. They can be either directed or not, allow multiple edges and loops, be weighted ore edge-labelled graphs etc. All depending on characteristics of described relations and how knowledge carried by such description is later used. Similarly to three approaches to groups identification also graph descriptions can be defined according to analogous premises. In this cases however, graphs of predefined interpretation or the one distinguished through supervised process seem to be more natural.

It is worth mentioning that presence of a bijective mapping between neurons and nodes from all graphs is meant. Thus sometimes we will refer to neurons as nodes, or even use set $V$ having set of nodes in mind. However, this is not a requirement.

### E. Dualism of inference process

During a network inference phase three processes are taking place in parallel. The first consists of neurons somatic parameters adaptation, namely reference points change or weight adaptation, network reorganisation, etc. The second process is an inference of neurons labelling, which covers label set and neuron membership management. The third, possibly the most challenging one, is inference of interrelation descriptions. This itself can be considered some kind of structural pattern analysis problem.

Although using the word “duality” in the following context may be seen as misuse, it intuitively describes character of relations between the inference of neural network and other two mentioned parallel processes. What is more, neuronal groups and interrelation graphs can be integral part of final result, not only the neural network structure and standard somatic parameters themselves.

## IV. Labelled Growing Neural Gas

To keep the presentation clear an example usage of *neuronal group* concepts will now be presented. Consequently, supervised single-label classification of $\mathcal{X} = \mathbb{R}^n$ elements will be analysed. To avoid misunderstanding, this problem can be defined as finding classifying function $h : \mathcal{X} \to \mathcal{L}$, where $\mathcal{L}$ denotes set of labels. In general $h$ of the form $h : \mathcal{X} \to 2^{\mathcal{L}}$ could be discussed (multi-label classification), but here we focus on simpler model for clarity and compactness reasons.

### A. Classifier model

Let $G = (V, E)$ be a strict (simple) graph, where $V$ and $E$ denote the node and the edge set respectively, and

$$\forall_{e \in E} \exists_{v_1, v_2 \in V} \; e = \{v_1, v_2\}$$

As it was suggested in section III we will consider $V$ as neuron set. Let each neuron $n \in V$ has associated element $w(n) \in \mathcal{X}$ here called node's *position* or *reference element*. Each $w(n)$ describes some or all somatic parameters (here weight vector) of a neuron associated with $n$.

Let $\tau : \mathcal{X} \to \mathbb{R}$ be a transmission function (composition of an *activity* and an *output* (activation) functions) of every neuron from $V$. We will restrict to $\tau$ being of the form $\tau = \sigma \circ \rho$, where $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is linear metric in $\mathcal{X}$, $\sigma$ is non-increasing function. These imply

$$\forall_{c \in V} \; \tau_n(\mathbf{x}) = \sigma(\rho(w(n), \mathbf{x})) \tag{2}$$

Metric can easily be exchanged with inner product without significant influence on other parts of presentation. In such case $\sigma$ should be non-decreasing function, as interpretation of inner product result is opposite to distance.

In addition, let $\mathbb{S} = \mathcal{L}$ and mutual group exclusion restriction (1) be imposed. To simplify notation $l(n)$ will denote the label of the group to which $n$ belongs (instead of a set having exactly one element- adequate group label).

Classifying function generated by such network can be defined in the following way:

$$\forall_{x \in \mathcal{X}} \; c_V(x) = l\left(\operatorname*{arg\,max}_{n \in V} \tau(x)\right) \tag{3}$$

It is worth explicit mentioning, that this model is equivalent to Nearest Neighbour algorithm, if one neuron would be constructed for every element $x$ in training set, with reference point and labelling set accordingly.

Graph $G$ will store topological neighbourhood interrelation information in context of input distribution of labelled points or training set. This will be the only description used in the prototype presented.

### B. Supervised Neuron Labelling

In the presented solution supervised neuron labelling is used. Policy must reflect the fact that it is not only used in learning phase (see section IV-C) but also for construction of final functional classifier hypothesis. The most straightforward and also effective supervised node labelling strategy is called *precision labelling*. It basically chooses the label that maximises hypothesis *precision* inside its *Voronoi* cell. Neurons *precision* is evaluated using training set or its subset/detached part. Obviously, *recall* and especially *F*-measures can be utilised as well.

*Activity labelling* can be seen as an alternative to *precision labelling*. While computing new label of neuron it takes also activity of neuron inside its *Voronoi* cell, choosing the *most active* group.

Labelling policy selection should reflect a general aim set to learning process. Referring to *Adaptive Active Hypercontour* techniques of classifier inference, an *energy* function can be used to estimate classifier quality during its inference. Let additionally $c_i$ be the classifier hypothesis in $i$-th classifier step, regardless step division meaning. The very simple energy function might be any measure of $h(c_i)$ *imprecision*. Minimising energy over training set is equivalent to maximising its precision. However, its true power depends on informative relation between training and test set. Other policies are possible and interesting, depending on entire process definition and its aims. The most promising one would be *contextual labelling*, taking advantage of hypothesis about labels of other nodes. Research about such technique is in progress.

Neuron labelling plays important role in adaptation phase, implying neurons reference points strategies. In this case, effective relabelling strategy is vital. Below the most notable approaches to this problem are listed.

1) Constant labelling—we assume that neuron's labelling is not altered during neural network learning. In this case overall algorithm efficiency depends on ability to generate new neurons belonging to a specific group. What is more, mislabelled node can be erased from $V$. In certain conditions, constant network with set labelling can also be used.
2) Off-line relabelling—in this case nodes' labellings do not follow dynamical changes in network structure and parametrisation. Relabelling is done using selected policy once for cone time, e.g. after each epoch (processing of entire training set).
3) On-line relabelling—where labelling policy is applied after every change in current process description.

Although on-line relabelling seems the most powerful and meaningful, it is also the most computationally expensive. Trade-off between effectiveness and efficiency of inference process should be analysed while planning to follow this way.

### C. Adaptation process

As mentioned earlier, the current labelling hypothesis can be used to drive network adaptation. A classic approach did not use such kind of information. Even considering *RBF network* setting, in which a sensor layer's aim is to decrease the space quantisation error rather than to find optimal quantisation for the second layer (learned in supervised process). One of problems addressed by this prototype implementation is to drive network inference in such a way, that would increase quantisation accuracy along surfaces dividing classes (areas surrounding contours).

Let the concept of *Adaptive Active Hypercontour* be presented in short, to build a frame for further considerations

Let $\mathcal{H}^{\mathcal{X},\mathcal{L}} = \{h : \mathcal{X} \to 2^{\mathcal{L}}\}$ denotes the set of (classifier) hypothesis over $\mathcal{X}$ and $\mathcal{L}$; it will be referred to as $\mathcal{H}$. As (in most cases) function realized by the classifier is considered *a solution*, a distinction is made between classifier $c \in C^{\mathcal{H}}$ referred as *hypothesis* and the classifying function $h(c) : \mathcal{X} \to 2^{\mathcal{L}}$ realized by $c$, which will be referred as *function hypothesis*.

Let $\kappa^p$ be parametrised by $p$ a task oriented classifier induction function, $C^{\mathcal{H}}_{\kappa}$ be classifier space (the domain of $\kappa$), such that $\forall_{c \in C^{\mathcal{H}}_{\kappa}} h(c) \in \mathcal{H}$. *AAH* solves classifier inference through optimisation process, which target function $E$ is commonly called *energy*. $E$ can be considered as function $\mathcal{H} \to \mathbb{R}$, but it is usually desired that it operates on $C^{\mathcal{H}}_{\kappa}$ directly, as various aspects of classifier parameters should be evaluated to achieve valuable solution.

*Adaptive Active Hypercontour* algorithm consists of the following steps [14]:

1) **Initialization**: during this phase the choice of initial classifier model and subordinate inference method ($\kappa$) is made. An initial classifier hypothesis $c_o \in C^{\mathcal{H}}_{\kappa}$ is then generated and its *energy* computed.
2) $\boldsymbol{\alpha - phase}$: This step mainly consists of subordinate algorithm invocation which generates new classifier hypothesis:
$$c_{i+1} = \kappa_p(c_i)$$
3) $c_{i+1}$ ***energy*** **estimation**
4) $\boldsymbol{\beta - phase}$: The subordinate algorithm ($\kappa$) is very often optimisation process itself. However, its target function usually is compatible, but different than $E$. In this phase **any** change in current process may be performed to ameliorate $E$ optimisation state modification, like:
   - unwarranted current hypothesis change, usually $\kappa$ and $E$ oriented, like restoring the previous classifier (in case of significant optimisation regression, or its structural change that bypasses $\kappa$ shortcomings).
   - $\kappa$ reparametrisation or replacement with different algorithm.
   - classifier model change.

The solution presented will now be described as an instantiation of presented meta-algorithm. Initialisation of the process consists of creating initial neurons with random reference points. Initial neuron labelling must also be performed; neuron interrelation graph has empty edge set $E$.

### D. Phase $\boldsymbol{\alpha}$

During $\alpha$ phase all inputs from training set or $\lambda$ random points generated according to training distribution are processed. On-line adaptation using all inputs is considered as one, atomic $\kappa$ invocation. For each input $\mathbf{x}$ having associated training label denoted as $l_{\mathbf{x}}$ the following steps are performed:

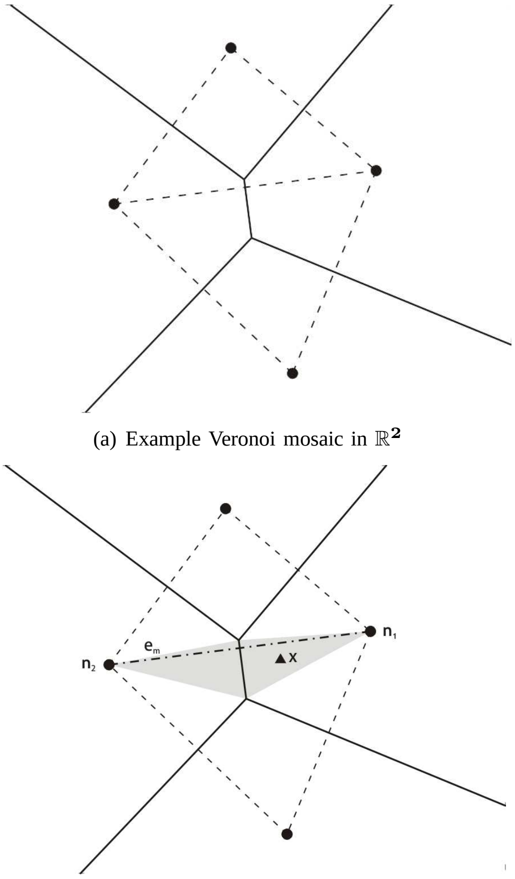
1) Find most active and second-most active neuron from $V$ for given $\mathbf{x}$, let them be $n_1$ and $n_2$ respectively.
2) Increase the age of all edges emanating from $n_1$ in $G$ by 1.
3) If $n_1$ and $n_2$ are connected with an edge $e_m$, set its age to zero. Otherwise, create edge $e_m = \{n_1, n_2\}$.
4) Apply neuronal position adaptation strategy. Few situations can be considered, presented here form "a straightforward example"
   - $\boldsymbol{l(n_1) \neq l(n_2)}$. In this case $e_m$ crosses the surface separating differently labelled areas of $\mathcal{X}$ according to the present function hypothesis. If one of the labels matches $l_{\mathbf{x}}$, then we perform a single-side surface approximation by adapting reference points of matching neuron using standard Kohonen rule:
   $$\Delta w_{n_1} = \alpha_1(\mathbf{x} - w(n_1)) \qquad (4)$$
   and its adjacent nodes in G using same rule with different, smaller step:
   $$\Delta w_a = \alpha_a(\mathbf{x} - w(n_1)) \qquad (5)$$
   If neither $l(n_1)$ nor $l(n_2)$ are equal to $l_{\mathbf{x}}$, then we perform adaptation of $n_1$ (together with its neighbourhood).
   - $\boldsymbol{l(n_1) = l(n_2)}$. In this case, $\mathbf{x}$ lays in "internal" region of label $l(n_1)$. If $l(n_1) = l_{\mathbf{x}}$, then no adaptation, or adaptation with negative steps should be performed. Negative learning is optional and, if enabled, should be used only when all neighbours of $n_1$ and $n_2$ have the same label equal to $l_{\mathbf{x}}$. If $l(n_1) \neq l_{\mathbf{x}}$ then standard adaptation of $n_1$ is applied.
5) Remove edges from $E$ with an age larger than $a_max$. If this results in neurons having no adjacencies—remove them from $V$.

To illustrate a typical situation during adaptation figures 1(a) and 1(b) are supplied, depicting a typical *Voronoi* cells structure and the active area of one of the edges.

### E. Phase $\boldsymbol{\beta}$

Once every phase $\alpha$, or rarely if desired, additional steps should be taken, e.g. to add new neurons to $V$. To enforce generation of new neurons in areas of our interest, namely along class discriminating surfaces, two following strategies are proposed.

**Intermediate node generation** is similar to the node generation mechanism used in *GNG* method. It can be done according to the subsequent steps:



(a) Example Veronoi mosaic in $\mathbb{R}^2$

(b) Example edge activity area

- Determine node $p$ of lowest precision in training set or randomly generated representative set of labelled inputs (if distribution is analysed).
- Insert new node $q$ with $w(q) = \frac{1}{2}(w(p) + w(r))$ where $r$ is neighbour of $p$ in $G$ having lowest precision.
- Insert edges $\{p, q\}$, $\{q, r\}$ into $E$ and remove edge $\{p, r\}$ from it.
- Relabel all neurons from $V$

This strategy requires global node relabelling. We can avoid it by using different policy of similar power called **node replication**. It simply replicates neuron of lowest precision, thus not altering Voronoi cell distribution.

### *F. Summary and possible extensions*

Presented *Labelled GNG* algorithm uses few concepts presented in previous section. At first, it uses supervised neuron labelling to infer classifying function of simple single-layer neural network. Although such network can be used as classifier (with formulae 3), its output can also be processed by other chained classifier model. In this case we will benefit from increased quantisation rate along decision surfaces.

The second important extension is *Neuronal Group Learning*, which utilises information about inferred *neuronal groups* during network adaptation process. This is used in both $\alpha$ and $\beta$ phases of the prototype method presented.

Empirical results and their analysis will be presented during the conference.

## V. NEURONAL INTERRELATIONS AS INFERRED KNOWLEDGE

The previous section presented method of *Labelled Growing Neural Gas* inference that can later be used as classifier. However, parallel inference of *neuronal groups* and *neuronal interrelation descriptions* can be the true problem solved by this method. At first, some other concept presented in [14] will be considered.

### *A. Neuronal Group Activity*

As suggested at the beginning, a formation of neuronal group does not have to be used only to facilitate learning process. Neurons can cooperate within one group and even compete with neurons from other groups during both learning and use stages.

This was first used in [14] in the concept of competitive learning of neural networks. Although differently described in original paper, let the concept of *neuronal group activity* be here introduced.

Let $\mathbb{F}$ be a common codomain of all neurons' transmission function, later referred as *field* or *output space*. In addition, let $\mathcal{L} \subset \mathbb{S}$, and $\forall(s \in \mathbb{S})V(s)$ denote all neurons from group $s$. Function hypothesis generated by neural networks with *neuronal group* information can be defined using the following formulae:

$$c(\mathbf{x}) = \arg\max_{l \in \mathcal{L}} \ \Psi_l(\mathbf{x}) \tag{6}$$

where $\Psi_l$ is activity of group $l$. It was proposed in [14] that $\Phi_l$ was of the form

$$\Psi_l(\mathbf{x}) = norm^{|V(l)|} \ (\Phi_{V(l)}(\mathbf{x})) \tag{7}$$

where $\Phi_{V(l)} : \mathcal{X} \to \mathbb{F}^{|V(l)|}$ be group activation function returning a vector of activations of neurons from specified group and $norm^k : \mathbb{F}^k \to \mathbb{R}$ be a proper norm (or semi-norm) in $\mathbb{F}^k$ space.

Interpreting set $\{(v, l(v)) : v \in V\}$ as a set of labelled control points, an appropriate definition of $\tau$ and $norm$ will result in classifier model equivalent to the one introduced be *potential method* and used in *Adaptive Potential Active Hypercontour* technique. Its properties allows efficient *energy* driven network inference, a solution to some optimisation problems faced during work presented in [9].

### *B. Structural Knowledge Acquisition*

One of the most promising fields of application of presented concepts are those connected with structure discovery and relation recognition, in which neural network is a backbone of *structural pattern analysis* process. Node distribution in space $\mathcal{X}$ induce some Voronoi cell structure. In described parallel processes relations among these cells can be recognised and inscribed in graphs of *neuronal interrelations*. Some of used graphs may have special meaning and can even form solution of *structural pattern analysis* problem.

To present an example, perspectives in the field of my special interest, data segmentation, will be analysed. Considering two dimensional segmentation problem (as image segmentation is) let neuron reference points lie in space $N \times N$ (as pixel coordinates do). It is quite natural, that neuron distribution in pixel coordinates space together with neuronal groups and labellings reflecting output labels forms a solution. However, additional somatic parameters and/or interrelation information may be needed during network learning.

Willing to provide argument proving potential of the concept presented it this field a straightforward image segmentation or contour inference algorithm can be defined in two simple ways.

The first uses concept of group activity and possibility to define them so as to get model equivalent *potential method*. In conjunction with meaningful definition of *energy* function, a *Potential Active Contours* are obtained. The other, better suited in neural network backbone of the method presented is known as *Kohonen Snakes*.

## VI. Conclusions and further work

The article presents *Neuronal Group* concept accompanied with based on graphs *neuron interrelation description* as a method of enriching neural network inference models. If properly defined *Neuronal Group Learning* can also lead to solution of *structural pattern analysis* tasks. Group descriptions and interrelations between neurons can explicitly be defined by a researcher. However, the widest perspectives are given by automatic, supervised or unsupervised interrelation inference.

Apart from extending existing methods, the one presented in this article can be used in advanced *structural pattern analysis* problems. Recent development and new tasks of *Information Retrieval* and *Text/Image Understanding* fields created a huge demand on contextual methods that would be flexible in utilisation of any kind of human knowledge about problem domain. What is more, automated knowledge inference from data is of increasing importance in all *Artificial Intelligence* domains. Future work on presented concepts will focus on these demanding task, especially that they proves to be much more difficult and requiring more sophisticated computational and analysis models than many currently considered.

## Acknowledgments

The author is grateful to Professor Piotr S. Szczepaniak from the Technical University of Lodz, Poland, for fruitful discussions.

## References

[1] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[2] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *Computational Learning Theory, Second European Conference, EuroCOLT ’95, Barcelona, Spain, March 13-15, 1995, Proceedings*, ser. LNCS, vol. 904. Springer, 1995, pp. 23–37.

[3] A. Esuli, T. Fagni, and F. Sebastiani, “MP-Boost: A multiple-pivot boosting algorithm and its application to text categorization,” in *Proceedings of the 13th International Symposium on String Processing and Information Retrieval (SPIRE’06), Glasgow, UK*, ser. LNCS, vol. 4209. Berlin / Heidelberg: Springer, 2006, pp. 1–12. [Online]. Available: http://www.isti.cnr.it/People/F.Sebastiani/Publications/SPIRE06a.pdf

[4] A. Tomczyk and P. S. Szczepaniak, “Contribution of active contour approach to image understanding,” in *Proceedings of IEEE International Workshop on Imaging Systems and Techniques—IST 2007, May 4–5, 2007, Krakow, Poland*, 2007.

[5] J. Wiebe, T. Wilson, and C. Cardie, “Annotating expressions of opinions and emotions in language,” *Language Resources and Evaluation*, vol. 39, no. 2, pp. 165–210, 2005.

[6] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data.” in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, 28 June – 1 July, 2001*, C. E. Brodley and A. P. Danyluk, Eds. Morgan Kaufmann, 2001, pp. 282–289.

[7] E. Breck, Y. Choi, and C. Cardie, “Identifying expressions of opinion in context.” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, 6-12 January 2007, Hyderabad, India,*, M. M. Veloso, Ed., AAAI Press, Menlo Park, California, 2007, pp. 2683–2688.

[8] A. Tomczyk and P. S. Szczepaniak, “Adaptive potential active hypercontours,” in *Proceedings of 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC). Zakopane. Poland*. Berlin. Heidelberg: Springer-Verlag, 2006, pp. 692–701.

[9] P. S. Szczepaniak, A. Tomczyk, and M. Pryczek, “Supervised web document classification using discrete transforms, active hypercontours and expert knowledge,” in *Proceedings of Web Intelligence Meets Brain Informatics, First WICI International Workshop, WImBI 2006, Beijing, China, December 15-16, 2006, Revised Selected and Invited Papers*, ser. LNCS, N. Zhong, J. Liu, Y. Yao, J.-L. Wu, S. Lu, and K. Li, Eds., vol. 4845. Springer, 2006, pp. 305–323.

[10] K. Ciesielski, “Adaptacyjne metody grupowania w mapowej wizualizacji kolekcji dokumentów tekstowych (eng: Adaptive grouping methods in map visualisation of document collection),” Ph.D. dissertation, Institute of Computer Science, Polish Academy od Sciences, Warszawa, 2007, (in polish).

[11] S. Haykin, *Neural Networks: A Comprechensive Foundation*. New York: Macmillan College Publishing Company, 1994.

[12] G. M. Edelman, *Neural Darwinism: The Theory of Neuronal Group Selection*. New York: Basic Books, 1987.

[13] M. Hadders-Algra, “The neuronal group selection theory: a framework to explain variation in normal motor development,” *Developmental Medicine & Child Neurology*, vol. 42, pp. 566–572, 2000.

[14] M. Pryczek, “Supervised object classification using adaptive active hypercontours with growing neural gas representation,” *Journal of Applied Computer Science*, 2008, (submited).

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Semantic Relevance Measure between Resources based on a Graph Structure

Sang Keun Rhee, Jihye Lee, Myon-Woong Park
Intelligence and Interaction Research Center
Korea Institute of Science and Technology
39-1 Hawolgok-dong, Seongbuk-gu
Seoul, Korea
Email: greyrhee@kist.re.kr

***Abstract*—Upon developing an information system, establishing an effective information retrieval method is the main problem along with the management of the information. With ontologies, the semantic structure of knowledge can be represented and the resources can be managed along with their context information. Observing that within this knowledge space, resources are not isolated but connected to each other by various types of relations, we believe that utilising those relations can aid information provisioning. In this paper, we explore the concept of the *semantic relevance* between resources and the semantic structure of a knowledge space, and present a relevance measuring algorithm between resources based on a graph structure, with two implementation examples.**

## I. Introduction

IN ANY knowledge space, each information resource does not exist in isolation but is connected to other resource(s) in many different types of relations. The relation may be explicitly represented or can be inferred, and resources can be directly linked or their relations may be indirect via another resource(s). Also, even two resources seem to have no relation whatsoever in a certain environment, they may have a contextual relation within another knowledge space. Concerning that the vast amount of readily available information in not only the Web but also in a limited knowledge space such as a single organisation, the information retrieval has been a challenging problem, and we believe that finding the semantic relevance between resources can contribute to effective information provisioning. To find the semantic relevance, the knowledge space needs to be semantically structured and represented, and a method in finding and measuring the relevance is also required.

In our earlier work [1], we presented our semantic relevance measure that calculates the relevance value between objects in a graph structure. In this paper, we enhance our initial approach by further discussing the meaning of the semantic relevance and describing the knowledge management within an information system with a knowledge model and an ontology, and creating the *Relevance Graph*, an interpretation of a knowledge model and the base structure of the relevance calculation, is explained in more detail. In addition, the relevance calculation algorithm is improved by adapting the *Edge Scaling* method presented in [2], and two systems which utilises this semantic relevance measure are introduced to describe examples of how this method can be implemented.

## II. Semantic Similarity and Semantic Relevance

There have been various different views and approaches to discover the similarity or relatedness among information objects. One of the most obvious relation that can be discovered between resources is the *semantic similarity*, which is the likeness between objects. For textual information such as documents, it means the similarity of their contents, and this similarity can be measured by lexical matching methods. One of the traditional methods is the TF-IDF(term frequency-inverse document frequency) [3] based comparison, and the LSA(latent semantic analysis) [4] provides an automatic way of organising documents into a semantic structure for information retrieval. The semantic similarity can also be found between terms as well as textual documents. Within a set of structured terms, as a tree-based hierarchy, several approaches have been introduced including an edge-based method [5] which measures the similarity from the number of edges between terms, a node-based methods [5], [6] that claim to be more accurate than the edge-based measure, and hybrid methods [7], [8] that overlay the edge-based approach with the node-based one.

From a different point of view, the similarity or relation between resources can be measured based on the links that connect the resources, and the *PageRank* [9] is possibly the most well-known link analysis algorithm. In [10], the links in the blogosphere are interpreted into an influence model and the *spreading activation technique* [11] is applied on the influence graph. Another approach is finding the relevance between objects based on tags by users, and a relevance measuring method based on tagging has been introduced in [12] for pictogram retrieval.

Now, let us describe our point of view on semantic relevance. We have discussed the semantic similarity between documents above, but sometimes it may be necessary to find more than the simple similarity of contents. Two documents can be related even though they do not contain similar contents or have links to each other, and such relation may be affect the resource ranking. For instance, suppose we have a document

This work is partially sponsored by the Korea Institute of Science and Technology (KIST) with the project TSI(Tangible Space Initiative).

describing an interface design methodology and another document explaining a data mining algorithm. In this case, these two documents are not considered to be similar, since their contents are about two different topics. However, if there is a research project developing a recommender system, where its user interface is designed following the methodology described in the first document and the recommendation algorithm is implemented based on the contents of the second document, then these two documents now have an indirect relation between each other within this project. Here, the fact that the methods described in both documents are utilised in the same project is the context information, and the two documents are related to each other in the given context although they are neither similar nor directly linked.

The semantic relevance can also be found between two objects which have completely different nature. For example, a person and a document are semantically related if the person is the author of the document. This can be further extended if the document has a topic and there exists a research project which is related to the topic, then in this case, a relation between the person and the project is discovered even if the person does not know the project exists.

Therefore, the *semantic relevance* we are exploring in this paper is based on any kinds of relations that can be represented in a knowledge space, and we are also considering indirect relations via context information. Now, to utilise such relevance in an information system, the degree of importance that each relation represents is considered as a numeric value, in other words, the closeness between two objects in the given knowledge space, and those values are then analysed to calculate the relevance value between any two objects. The relevance value calculation algorithm is described later in this paper.

## III. Semantic Knowledge Management

### A. Semantic Knowledge Space

Within an information system, there exists a set of information, and not only the information objects but also the relations among them are important for effective information provisioning. These information objects and their relations form a *knowledge space*, and its structure needs to be represented semantically to build a semantic knowledge space. The simplest form of a knowledge space consists of a set of information (or data) that is to be provided to users - in other words, *resources* - without any semantic annotations. Then, links between those resources can be included in the knowledge space to represent relations among those resources. An intuitive example of this would be a set of documents where each document contains links to other document(s). In this knowledge space, the connections between the resources are described, but this structure still does not have semantic meanings as the relations are represented only via *syntactic links*. To describe the semantics of the resources, the meaning of each relation needs to be assigned to those links so that they become *semantic links*. In addition to those semantically linked resources, context information can be added to the knowledge space to represent richer semantic meanings of those resources within a certain context.

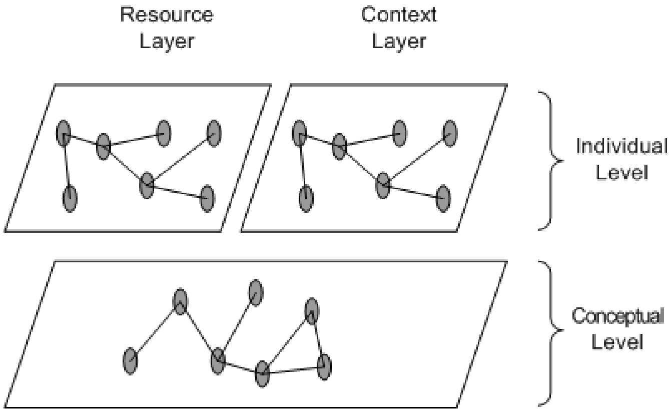


Fig. 1. Knowledge Model

Therefore, a *semantic knowledge space* is a semantic structure of information which contains resources, context information, and relations between them where each relation has its own meaning. We will now describe the structure of a semantic knowledge space in detail with a knowledge model and an ontological representation.

### B. Knowledge Model and Ontology

In our point of view, a semantic knowledge space consists of two types of information, context information and resources. As discussed above, resources are the information objects that are to be provided to users, and context information is the additional environmental or domain information. For effective knowledge management, before describing each individual information objects and their relations, a structure of knowledge space needs to be designed in conceptual level first, then the individual information objects can be represented within the conceptual structure. To represent such semantic knowledge, ontology provides a suitable formal structure, and we will now discuss the ontological representation of a knowledge space along with a knowledge model view, with a simple example ontology of a research organisation. Note that the example ontology presented in this section is severely simplified mainly to support explaining the knowledge structure and relevance calculation. This example will be used throughout the rest of this paper.

The structure of our knowledge model is shown in Fig. 1 to describe the overview of a semantic knowledge space from our point of view. First, in the *Conceptual Level*, the structure of concepts, the relations between them, and additional attributes are defined. Suppose we want to manage and provide information on people and documents within a research organisation, then four concepts may be defined in this level, *Project*, *Topic*, *Person*, *Document*, and the relations between these concepts can also be specified. Therefore, the *Conceptual Level* can be seen as a base structure of an ontology specifying the concepts and their properties, without individual instances. Fig. 2 presents our example ontology.

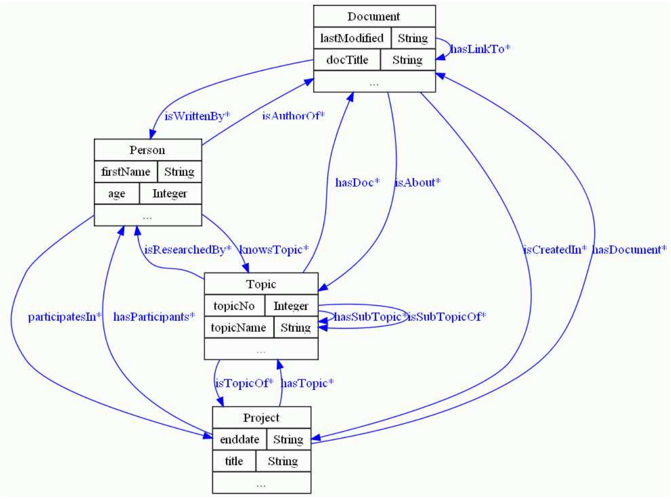


Fig. 2. An example ontology

The *Individual Level* contains the resources and context information represented as instances of the ontology defined in *Concept Level*, and it is divided into two layers - the *Resource Layer* and the *Context Layer*. The *Resource Layer* contains the information to be provided to users, and in our example, a set of documents and a set of people can be regarded as the contents of this layer. The *Context Layer* contains the domain and context knowledge, and a hierarchy of topics and a list of projects are stored in this layer in our example. The contents of these layers are semantically linked to each other based on the relations defined in the *Conceptual Level*.

## IV. Relevance Graph

Having the structured knowledge space, our next objective is measuring the semantic relevance between resources, represented in the *Resource Layer*. To discover their relevance, we interpret the knowledge space (i.e. ontology) into a graph structure where the relevance can be calculated as a numerical value. A *Relevance Graph* is an interpreted semantic knowledge model based on a knowledge structure, or ontology, representing the information objects and their relations. It is defined as a directed labelled graph $G = (V, E)$ where:

- $V$ is a set of nodes, representing individuals;
- $E$ is a set of edges, representing relations between individuals.

Note that our *Relevance Graph* does not have any restrictions in its structure. It is not limited to a tree structure, and different edges can represent different relation types. There may be multiple edges between two adjacent nodes, and the graph can contain cycles. Also, depending on each implementation and their purposes, a whole ontology can be interpreted into a graph, or only a part of an ontology can become the relevance graph. Or, even multiple ontologies can form a single relevance graph, although an ontology matching would be required in this case, and it is not discussed in this paper. The creation of the *Relevance Graph* includes node creation, edge creation, and edge labelling.

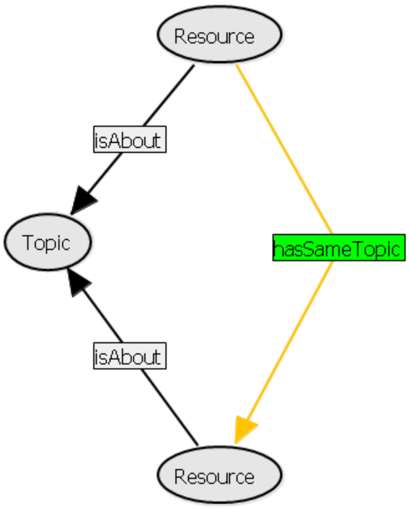


Fig. 3. An example rule

### A. Node Creation

There can be two different ways of creating nodes in our *Relevance Graph*. First, all instances of an ontology can become nodes, or second, only the instances representing *resources* can be created as nodes. The former is simpler way, but the result graph become bigger so it requires more computational time in the relevance calculation phase. The latter requires defining additional rules to clarify the relations between resources via context information since those relations may disappear upon graph creation as individuals representing context information is not included in the graph. For example, upon creating the nodes based on the example ontology described in the previous section (Fig. 2), each instance representing a `Document` or a `Person` becomes each node, but the individuals of `Topic` or `Project` do not appear on the graph, hence indirect relations between resources via these two conceptual information can be lost. Therefore, additional rules need to be defined to keep the meaningful indirect relations, and one example rule may be representing ‘a document is about the same topic as another document.’ This example rule is shown in Fig. 3 and this can be represented in SWRL [13] as follows:

```
<ruleml:imp>
  <ruleml:_rlab  ruleml:href="#sametopic"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom
      swrlx:property="isAbout">
       <ruleml:var>d1</ruleml:var>
       <ruleml:var>t1</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom
      swrlx:property="isAbout">
       <ruleml:var>d2</ruleml:var>
       <ruleml:var>t1</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom
      swrlx:property="hasSameTopic">
```

```
        <ruleml:var>d1</ruleml:var>
        <ruleml:var>d2</ruleml:var>
      </swrlx:individualPropertyAtom>
    </ruleml:_head>
  </ruleml:imp>
```

However, finding all meaningful indirect relations between resources and defining them as rules may not be a trivial task, hence in practice, the recommendable method is combining both ways of node creation depending on the characteristics of each application—creating all resources and 'some' context instances as nodes, and adding additional rules.

### B. Edge Creation

The next step is creating edges between the nodes in the *Relevance Graph*. The edges represent the relations between nodes, and they can be created from the relations (or object properties in OWL) defined in the ontology. Not only explicit relations but also inferred ones between resources by the additional rules become edges. Depending on the knowledge structure and application, all such relations in the ontology can become edges or only the relations which represent meaningful relevance in the system can be selected. Note that any reflexive relations are ignored hence do not become edges, since they have no significance in measuring relevance between nodes:

$$\forall x \in V : \qquad e(x,x) \notin E$$

### C. Edge Labelling

The edges in our *Relevance Graph* represent various relations defined in the ontology, and each relation has different importance in terms of representing the 'closeness' or 'relevance' between objects within a knowledge space. For instance, if we have two relations `hasSameTopic` and `hasSameAuthor` representing relations between documents, `hasSameTopic` can be regarded to be more important than `hasSameAuthor` because having the same topic means the contents of the two documents are very likely to have a close relationship whereas an author may write two different articles in two completely different area. Based on this assumption, each relation on the ontology can be weighted with a different value representing its importance within the system, and these values become the labels of the edges in the *Relevance Graph*. Note that, the value we are assigning here is the 'distance' value, which is defined as the inverse of the relevance value:

$$Distance = \frac{1}{Relevance} \tag{1}$$

Upon implementation, the distance value assignment can be included in the ontology (as in [14] and [2]) or in the graph creation module (as in [15]). Currently, we do not have an automated way of assigning the relevance value to each relation, so it needs to be done manually by the ontology developer and/or a domain expert. Practically, all distances can be instantiated with a single value, and the value for each relation can then be updated by applying the developers' domain knowledge and throughout testing, which was the initial implementation approach in [15] and was also followed in [2]. Also, there can be a (semi-)automatic way of updating the distance value via historical evaluation methods based on the system usage and explicit/implicit feedback, and this will be explored and experimented further in our future work.

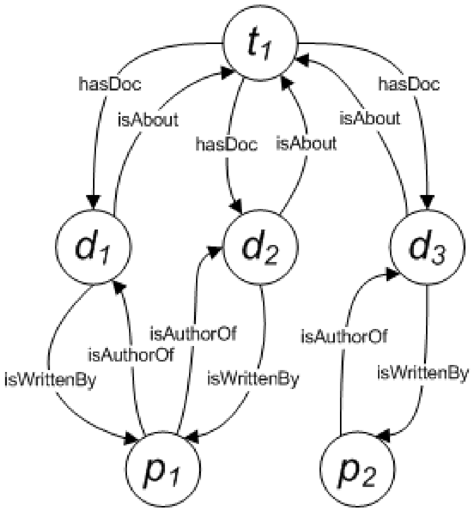


Fig. 4. An example illustration of relations among instances

## V. Relevance Calculation

Having created the *Relevance Graph*, the relevance value between two nodes in the graph can now be calculated by a relevance calculation algorithm, which is based on our earlier work [1]. Before we proceed further, let us elucidate the assumptions of our algorithm and define the terms to clarify their meanings and avoid possible confusion. The two fundamental assumptions are as follows:

1) Each relation has different importance, and it can be represented as a numeric value.
2) Having more relations between two resources means that they are closer(more relevant).

The assumption 1) is already discussed in Section IV-C, and the assumption 2) can be explained with an example case presented in Fig. 4. In this figure, we have a graphical representation of relations among a topic($t_1$), persons($p_1$, $p_2$), and documents($d_1$, $d_2$, $d_3$). Considering the relevance between documents $d_1$ and $d_2$, and between $d_2$ and $d_3$, it is safe to say that—it may not be true in all cases but has high probability—$d_1$ is likely to be more relevant to $d_2$ than $d_3$ is, as $d_1$ has the same topic($t_1$) and the same author($p_1$) as $d_2$ while $d_3$ has the same topic($t_1$) but a different author($p_2$). Based on this assumption, the merging algorithm is developed instead of taking a shortest(or longest) edge(or path) or taking the mean value (see Section V-B and V-D).

The terms used in the rest of this paper is defined as follows:

- *Relevance* represents the closeness between two entities. Each relation has its own relevance value.
- *Distance* represents how far two concepts are away from each other. The distance value is the inverse of the relevance value (see equation (1)).

- *Weight* represents the closeness between two entities in individual level. The weight value (0,1) is used to scale the distance (or relevance) value appropriately to each individual.

Now, the relevance values between the nodes in our *Relevance Graph* are measured by the following four steps.

### A. Edge Scaling

The first step is individualising each edge's label (i.e. distance). It was not included in our original work [1] but developed while implementing the relevance calculation method in a *duty-trip support system* [2]. The edge label is set based on the importance of a certain relation in comparison to other relations in conceptual level(see Section IV-C), hence all instances connected by the same relation have the same distance value. However, depending on the application and its knowledge structure, it might be desired to represent the different degree of relevance to certain relations in individual level. For example, in Fig. 2, we defined the `isAbout` relation that connects `Document` and `Topic`, and the `isAbout` relation has a certain *distance* value. However, there can be two different documents related to the same topic but with different degree. Here, the degree of relevance can be defined in the ontology as following:

```
onto:Doc1
  a onto:Document ;
  onto:isAbout
    [ a onto:DocTopic ;
      onto:relatedTopic onto:Ontology ;
      onto:weight ”0.90”^^xsd:float
    ] .

onto:Doc2
  a onto:Document ;
  onto:isAbout
    [ a onto:DocTopic ;
      onto:relatedTopic onto:Ontology ;
      onto:weight ”0.25”^^xsd:float
    ] ;
  onto:isAbout
    [ a onto:DocTopic ;
      onto:relatedTopic onto:Java ;
      onto:weight ”0.40”^^xsd:float
    ] .
```

The above example presents two documents `Doc1` and `Doc2` which are both related to a topic `Ontology` with different weight values. To apply this in our relevance graph, we multiply the initial *relevance value* by this *weight value* and obtain the scaled distance value.

$$newRelevance = relevance \times weight \tag{2}$$

Therefore, from the equation (1), for an edge $e(x,y) \in E$ where $x$ and $y$ are two adjacent nodes $x, y \in V$, its initial distance value $D_{e(x,y)}$, and the weight value $w_{xy}$, the scaled distance value $D'_{e(x,y)}$ is:

$$D'_{e(x,y)} = (\frac{1}{D_{e(x,y)}} \times w_{xy})^{-1} \tag{3}$$

We can regard this process as *individualisation* of the edge distance since the initial distance was determined in conceptual level and it is scaled by the weight value assigned in individual level. This step is optional part in relevance calculation since it can only be applied when the *weight* values are defined between instances in the knowledge structure (i.e. ontology).

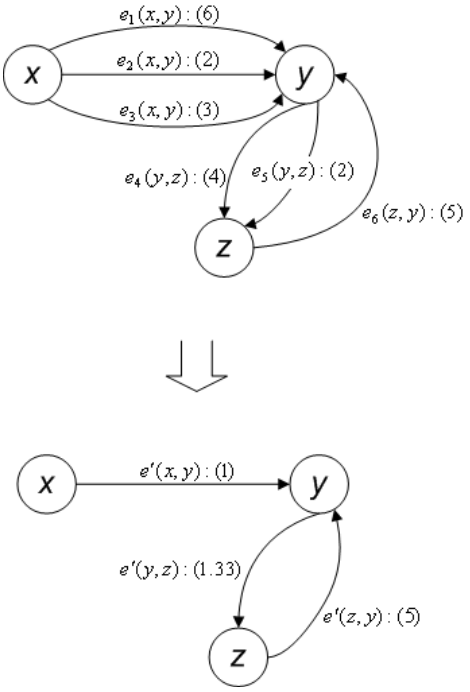


Fig. 5. Edge Merging

### B. Edge Merging

In our *Relevance Graph*, there may be multiple edges with the same direction between two adjacent nodes, since two resources can be connected via more than one relation. Therefore, from our second assumption of the algorithm described at the beginning of this section, we merge those edges into a single one so that we can obtain a simpler graph where there is only a single edge with the same direction between two adjacent nodes. To calculate the distance value of the merged edge, we first calculate the relevance between two adjacent nodes. For two adjacent nodes $x, y \in V$, and edges $e_1(x,y), e_2(x,y), \ldots, e_n(x,y) \in E$ from node $x$ to node $y$ with the distance value $D_{e_i(x,y)}$ for edge $e_i(x,y)$ where $1 \leq i \leq n$, the semantic relevance value $r_{xy}$ from node $x$ to node $y$ by direct relations(edges) is as follows:

$$r_{xy} = \sum_{i=1}^{n} \frac{1}{D_{e_i(x,y)}} \tag{4}$$

Therefore, from the equation (1), the new distance value $D_{e'(x,y)}$ of the merged edge $e'(x,y)$ is:

$$D_{e'(x,y)} = \frac{1}{r_{xy}} = (\sum_{i=1}^{n} \frac{1}{D_{e_i(x,y)}})^{-1} \tag{5}$$

By merging edges, we can now obtain a simpler graph where there is only a single edge with the same direction between adjacent nodes. Fig. 5 describes an example illustration of edge merging.

### C. Path Distance Calculation

Now we need to consider the relevance between non-adjacent nodes. For two non-adjacent nodes, there may or may not exist a path. In our algorithm, a path is valid if and only if it contains no repeated nodes (i.e. simple path). If there does not exist a path between them, it means that they are not related and we can consider their relevance value as $0$. If there exist one or more paths, we first need to calculate the distance value of each path.

In graph theory, the weight(or distance) of a path in a weighted graph is the sum of the weights of each edge in the path. Following this, for a path $P(a_1a_n)$ that visits $n$ nodes $a_1, a_2, \ldots, a_n \in V$, the path distance $D_{P(a_1a_n)}$ is:

$$D_{P(a_1a_n)} = \sum_{k=1}^{n-1} D_{e'(a_ka_{k+1})} \tag{6}$$

However, instead of selecting a single path, we consider all relations in our relevance calculation, and it is discovered that this can produce undesirably close relevance results for between two nodes which are connected via long paths. Precisely, from a node $x$, a node $y$ should be closer than a node $z$ but the result can be the opposite because of the number of paths from $x$ to $z$ even though each path is very long. Hence, it is necessary to make the distance value between indirectly related nodes higher than the simple sum, and this is done by multiplying the edge count $k$ to the distance value of each edge. Therefore, the above equation (6) is replaced with the following:

$$D_{P(a_1a_n)} = \sum_{k=1}^{n-1} (k \times D_{e'(a_ka_{k+1})}) \tag{7}$$

### D. Path Merging

The last step of the relevance calculation is handling multiple paths between two nodes, and this is done by following the same principle as edge merging. For $n$ paths $P_1, P_2, \ldots, P_n$ from node $x \in V$ to node $y \in V$, the relevance value $R_{xy}$ is:

$$R_{xy} = \sum_{k=1}^{n} \frac{1}{D_{P_k(xy)}} \tag{8}$$

$R_{xy}$ represents the final semantic relevance value of node $y$ from node $x$.

## VI. Implementation Examples

The semantic relevance measure introduced in this paper has been implemented in two systems—RIKI and SPIP.

### A. RIKI

RIKI[15] is a Wiki-based portal supporting group communication and knowledge sharing within a collaborative research project. Unlike other *Semantic Wikis*[16][17][18], ontological knowledge management is applied in RIKI mainly to provide easier knowledge access within a specific project's context, rather than rich semantic annotation. Its knowledge space is developed following the knowledge model and structure introduced in this paper (Section III), and the knowledge space is structured in F-Logic[19] ontology. Two top-level concepts are defined in this ontology, the `Resource` containing the Wiki articles and the `Context` representing the context information. These are further divided into four sub-concepts, classifying the article types and describing different types of the contexts. The overview of this structure is depicted in Fig. 6.

RIKI provides two ways of navigating the articles—the *Structured Browsing* and the *Article Recommendation*. The *Structured Browsing* provides a tree-based structure of the contexts, so that users can see the overall structure of the information in RIKI and reach the desired article(s) related to a certain topic, task, event, or person. By the *Article Recommendation* function, while a user is viewing an article, the system generates and displays a list of its relevant articles. To create the *Relevance Graph* from the ontology, first, all the instances of the `Resource` concept and its subconcepts (i.e. articles) are created as nodes. Then, for edge creation, we defined 50 rules representing the indirect relations between those articles via the given contexts, and those rules became edges of our graph. From this *Relevance Graph*, when a user opens an article, the system finds the top 10 relevant articles from the current article by the algorithm presented in this paper. The *Edge Scaling* part is not included in this RIKI implementation. Fig. 7 presents the RIKI interface, with the *Structured Browsing* on the top-left and the *Article Recommendation* on the bottom-left. The details of the overall system can be found in [15].

### B. Smart Personalized Information Provider

Another implementation example of the semantic relevance measure is the *Smart Personalized Information Provider*, which is sponsored by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant. It aims for adaptive information provisioning in an agent-based virtual organisation, and provides a personalised information by ontological resource matching. The knowledge space is developed in OWL (details in [20]), and it provides two different services—the *Grant Announcement Service* and the *Duty Trip Support*. In both services, for resource matching, the relevance calculation algorithm is utilised, along with a SPARQL[21] engine for resource filtering and the *GIS Subsystem* for geospacial information management. The matching process is described in detail in [14] for the *Grant Announcement Service* and also in [2] for the *Duty Trip Support*.

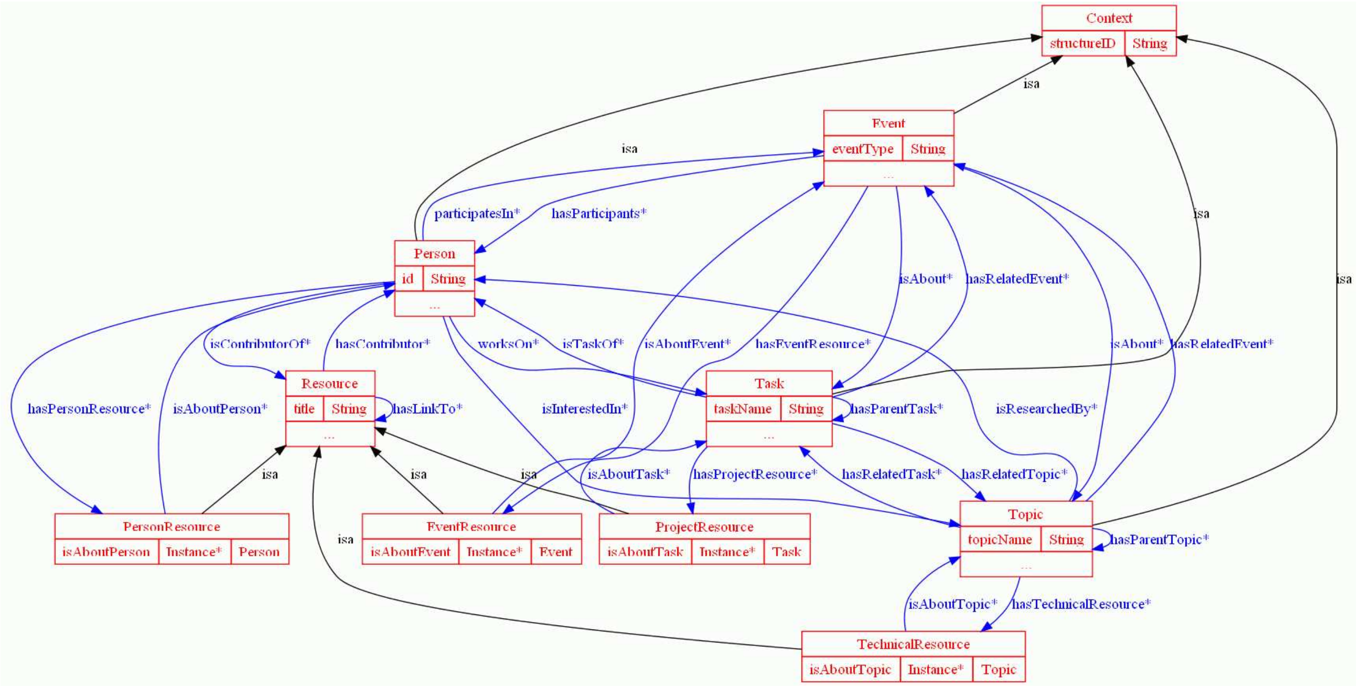
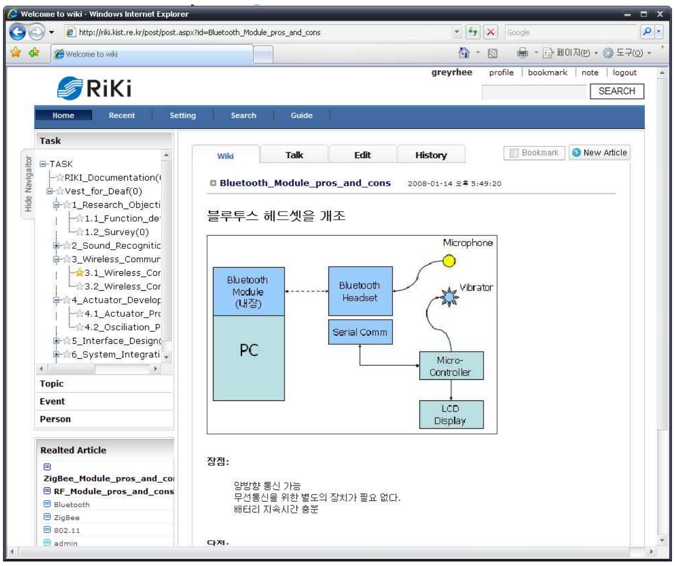


Fig. 6. The RIKI ontology.



Bluetooth_Module_pros_and_cons 2008-01-14 오후 5:49:20

블루투스 헤드셋을 개조

장점:

양방향 통신 가능
무선통신을 위한 별도의 장치가 필요 없다.
배터리 지속시간 충분

단점:

Fig. 7. The RIKI interface.

## VII. Concluding Remarks

In this paper, we discussed our view of the concept *semantic relevance* and presented a structured view of a semantic knowledge space, with the semantic relevance measure algorithm between the resources based on a graph structure. The relevance measure algorithm can be implemented on its own, but the efficiency will be more evident when it is combined with existing searching or matching algorithms, such as the text-based comparing methods and/or tagging-based relevance measure. Our future work will include the integration of our method with those other techniques, as well as exploring a semi-automatic method of initiating and updating the edge labels, experimenting with various implementations and evaluating throughout extensive testing.

## References

[1] S. K. Rhee, J. Lee, and M.-W. Park, "Ontology-based Semantic Relevance Measure," *in Proceedings of the 1st SWW Workshop(ISWC)*, Korea, 2007.

[2] M. Szymczak, G. Frackowiak, M. Ganzha, M. Paprzycki, S. K. Rhee, J. Lee, Y. T. Sohn, J. K. Kim, Y.-S. Han, and M.-W. Park, "Ontological Matchmaking in an Duty Trip Support Application in a Virtual Organization," *in Proceedings of the IMCSIT'08 Conference*, 2008, in press.

[3] R. R. Korfhage, *Information Storage and Retrieval*, Wiley, 1997.

[4] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and K. Harshman, "Using latent semantic analysis to improve access to textual information," *in Proceedings of the CHI'88 Conference*, ACM Press, 1988, pp. 281–283.

[5] P. Resnic, "Using Information Content to Evaluate Semantic Similarity in a Texonomy," *IJCAI*, 1995, pp. 448–453.

[6] D. Lin, "An Information-Theoretic Definition of Similarity," *in 15th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 1988, pp.296–304.

[7] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," *in Proceedings of the International Conference on Research on Computational Linguistics*, Taiwan, 1997.

[8] C. Leacock and M. Chodorow, "Combining Local Context and WordNet Similarity for Word Sense Indentification," *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998, pp. 265–283.

[9] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *in Proceedings of the 7th WWW Conference*, Brisbane, Australia, 1998.

[10] A. Java, P. Kolari, T. Finin, and T. Oates, "Modeling the Spread of Influence on the Blogosphere," *in Proceedings of the 15th WWW Conference*, 2006.

[11] F. Crestani, *Application of Spreading Activation Techniques in Information Retrieval*, Artificial Intelligence Review, 1997.

[12] H. Cho, T. Ishida, R. Inaba, T. Takasaki, and Y. Mori, “Pictogram Retrieval Based on Collective Semantics,” *in Proceedings of the 12th HCI Conference,* LNCS 4552, 2007, pp. 31–39.

[13] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/.

[14] M. Szymczak, G. Frackowiak, M. Ganzha, M. Paprzycki, S. K. Rhee, J. Lee, Y. T. Sohn, J. K. Kim, Y.-S. Han, and M.-W. Park, “Infrastructure for Ontological Resource Matching in a Virtual Organization,” *in Proceedings of the IDC’2008 Conference,* Studies in Computational Intelligence, N. Nguyen and R. Katarzyniak, Eds., vol. 134, Heidelberg, Germany: Springer, 2008, pp. 111–120.

[15] S. K. Rhee, J. Lee, and M.-W. Park, “Riki: A Wiki-based Knowledge Sharing System for Collaborative Research Projects,” *in Proceedings of the APCHI 2008 Conference,* LNCS 5068. Springer, 2008.

[16] M. Krotzsch, D. Vrandecic, M. Volkel, H. Haller, R. Studer, “Semantic MediaWiki,” *Journal of Web Semantics,* 2007, pp. 251–261.

[17] E. Oren, “SemperWiki: A Semantic Personal Wiki,” *in Proceedings of the Semantic Desktop Workshop,* 2005.

[18] S. Schaffert, “IkeWiki: A Semantic Wiki for Collaborative Knowledge Management,” *in Proceedings of the 1st International Workshop on Semantic Technologies in Collaborative Applications,* 2006.

[19] M. Kifer, G. Lausen, and J. Wu, “Logical Foundations of Object-Oriented and Frame-Based Languages,” *Journal of ACM,* 1995.

[20] M. SzymczakMichal, et al.M. Szymczak, G. Frackowiak, M. Gawinecki, M. Ganzha, M. Paprzycki, M.-W. Park, Y.-S. Han, and Y. Sohn, “Adaptive Information Provisioning in an Agent-Based Virtual Organization – Ontologies in the System,” *in Proceedings of the KES-AMSTA Converence,* LNAI, N. Nguyen, Ed., vol. 4953. Heidelberg, Germany: Springer, 2008, pp. 271–280.

[21] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 237–242

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Exclusion Rule-based Systems—case study

Marcin Szpyrka
AGH University of Science and Technology
Department of Automatics, Kraków, Poland
Jan Kochanowski University
Institute of Physics, Kielce, Poland
Email: mszpyrka@agh.edu.pl

***Abstract*—The *exclusion rule-based system*, proposed in the paper, is an alternative method of designing a rule-based system for an expert or a control system. The starting point of the approach is the set of all possible decisions the considered system can generate. Then, on the basis of the exclusion rules, the set of decisions is limited to the ones permissible in the current state.**

**A railway traffic management system case study is used in the paper to demonstrate the advantages of the new approach. To compare the exclusion rule-based systems with the decision tables both solutions are considered.**

## I. Introduction

ALTHOUGH rule-based systems are widely used in various kinds of computer systems, e.g. expert systems, decision support systems, monitoring and control systems, diagnostic systems etc., their encoding, analysis and design can be a time-consuming, tedious and difficult task ([1], [2], [3], [4]). Their expressive power, the scope of the potential application and very intuitive form, make them a very handy and useful mechanism. However, if more complex systems are considered, it is difficult to cope with the number of attributes or/and the number of decision rules.

Rule-based systems can be represented in various forms, e.g. decision tables, decision trees, extended tabular trees (XTT, [5]), Petri nets ([6]), etc. Decision tables seem to be the most popular form of rule-based systems presentation. Depending on the way the condition and decision entries are represented, a few kinds of decision tables can be distinguished ([7], [8], [3]). The simpler ones are decision tables with atomic values of attributes. However, encoding decision tables with the use of atomic values of attributes only is not sufficient for many real applications. On the other hand, tables with generalised decision rules ([9]) use formulae instead of atomic values of attributes. Each generalised decision rule covers a set of decision rules with atomic values of attributes (simple decision rules). Therefore, the number of generalised decision rules is significantly lower than the number of the simple ones.

In spite of this, it can be really hard to cope with the design of such a decision table if it contains a dozen or so condition attributes or/and each of them can take a few different values. It appears that it is simpler to point out one or two values of some condition attributes that disqualify some decisions, than to point out values of dozen attributes that determine a single decision.

This observation provides the basis for the approach presented in the paper. We start with the set of all possible decisions the considered system can generate. Then, instead of constructing the precondition of a decision rule that usually contains most of the condition attributes, we construct the precondition of an exclusion decision rule that usually contains a single attribute. Thus, instead of pointing out a single decision permissible in the current state, we point out a set of impermissible ones.

In most basic versions, a rule-based system consists of a single-layer set of rules and a simple inference engine. It works by selecting and executing a single rule at a time, provided that the preconditions of the rule are satisfied in the current state. Thus, such a system generates a single decision, even if there are a few suitable ones. On the other hand, when an exclusion rule-based system is used, then executing a single exclusion rule removes some impermissible decision from the set of all possible ones. After executing all exclusion rules with satisfied preconditions, the set contains all decisions permissible in the considered state.

The paper presents the idea of *exclusion rule-based systems* (shortly ExRBS). To point out the main differences between generalised decision tables and exclusion rule-based systems, we describe the two forms of rule-based systems for a railway traffic management system. The system uses a rule-based system to choose routes for trains moving through a train station.

The paper is organized as follows. The railway traffic management system is described in Section II. The generalized decision table is presented in Section III. The exclusion rule-based system is described in Section IV. The paper ends with a short summary in the final section.

## II. Railway traffic management system

A description of a railway traffic management system for a real train station is discussed in this section. To present complete rule-based systems, a small train station, Czarna Tarnowska, belonging to the Polish railway line no 91 from Kraków to Medyka, has been chosen. This example seems to be suitable for the presentation of an exclusion rule-based system. However, it will be shown that the approach can also be applied to more complex systems.

Acknowledgments: The research presented in this paper is supported from MNiSW Research Project HeKatE No.: N516 024 32/2878, Agreement No.: 2878/T02/2007/32.

TABLE I
REQUIRED POSITION OF TURNOUTS FOR ALL POSSIBLE ROUTES

| | Turnouts | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Routes | 3/4 | 5 | 6 | 7/8 | 15/16 | 17 | 18 | 19/20 | 21/22 |
| B1 | + | + | | | | | | | |
| B2 | – | | + | o+ | | | | | |
| B3 | + | – | | | | | | | |
| B4 | – | | – | + | o+ | | | | |
| R2 | | | | o+ | o+ | + | | + | + |
| R4 | | | | o+ | + | – | | + | + |
| F2W | + | | + | o+ | | | | | |
| G2W | + | | – | + | | | | | |
| K1D | | | | | + | – | | – | + |
| L1D | | | | | o+ | + | | – | + |
| M1D | | | | | | | + | + | + |
| N1D | | | | | | | – | + | + |

TABLE II
RELATIONSHIPS BETWEEN ROUTES

| | B1 | B2 | B3 | B4 | R2 | R4 | F2W | G2W | K1D | L1D | M1D | N1D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B1 | – | x | x | x | | | | | | | | xx |
| B2 | x | – | x | x | xx | | x | x | | | | |
| B3 | x | x | – | x | | | | | | | | |
| B4 | x | x | x | – | xx | xx | x | x | | | | |
| R2 | | xx | | xx | – | x | | xx | x | x | | |
| R4 | | | | xx | x | – | | | x | x | | |
| F2W | | x | | x | | | – | x | | | | |
| G2W | | x | | x | xx | | x | – | | | | |
| K1D | | | | | x | x | | | – | x | x | x |
| L1D | | | | | x | x | | | x | – | x | x |
| M1D | | | | | | | | | x | x | – | x |
| N1D | xx | | | | | | | | x | x | x | – |

The considered system is used to ensure safe riding of trains through the station. It collects some information about current railway traffic and uses a rule-based system to choose routes for trains.

The topology of the train station with original signs is shown in Fig. 1. The letters A, B, D, etc. stand for color light signals, the symbols Z3, Z4, Z5, etc. stand for turnouts and JTA, JTB, JT1, etc. stand for track segments. Some simplifications have been introduced to reduce the size of the model. We are not interested in controlling the local shunts so the track segment JT6 will not be considered. We assume that the light signals display only two signals: *stop* and *way free*. Moreover, outside the station the trains can ride using the right track only.

A train can ride through the station only if a suitable route has been prepared for it i.e., suitable track segments must be free, we have to set turnouts and light signals and to guarantee exclusive rights to these elements for the train. Required position of turnouts for all possible routes are shown in Table I, where the used symbols stand for:

- + closed turnout (the straight route);
- – open turnout (the diverging route);
- o+ closed turnout (for safety reasons).

For example, the symbol B4 stands for the input route from the light signal B to the track no. 4. The symbol F2W stands for the output route from the track no. 2 (from the light signal F) to the right (to Wola Rzędzińska), etc. The route B4 can be used by a train only if: turnouts 7, 8, 15, 16 are closed, turnouts 3, 4, 6 are open, and the track segments JTB, JT4, JZ4/6 (a segment between turnouts 4 and 6), JZ7 (diagonal segment leading to the turnout 7) and JZ16 are free.

Some routes cannot be set at the same time because of different position of turnouts or for safety reasons. Mutually exclusive routes are presented in Table. II, where the used symbols stand for:

- x – mutually exclusive (different position of turnouts);
- xx – mutually exclusive (safety reasons).

The system is expected to choose suitable routes for moving trains. It should take under consideration that some trains should stop at the platform, while others are only moving through the station and two routes (an input and an output one) should be prepared for them. In such a case, if it is not possible to prepare two routes, only an input one should be prepared.

## III. DECISION TABLE

Decision tables are a precise and compact form of rule-based systems presentation. They vary in the way the condition and decision entries are represented. The entries can take the form of simple true/false values, atomic values of different types, non-atomic values or even fuzzy logic formulas [3].

To construct a decision table, we draw a column for each condition (or stimulus) that will be used in the process of taking a decision. Next, we add columns for each action (or response) that we may want the system to perform (or generate). Then, for every possible combination of values of those conditions a row should be drawn. We fill cells so as to reflect which actions should be performed for each combination of conditions. Each such a row is called a *decision rule*.

A kind of decision tables with generalized rules is used in this section [9]. Systems with non-atomic values are considered since their expressive power is much higher than the one of classical attributive decision tables. Each cell of such a decision table should contain a formula which evaluates to a boolean value for condition attributes, and to a single value (that belongs to the corresponding domain) for decision attributes.

Let us focus on the railway traffic management system. The rule-based system is to be used to determine which routes should be prepared depending on the data collected from sensors. The decision table contains 20 condition and 2 decision attributes. The condition attributes stand for information about:

- current position of the train (attribute JT) – before the light signal B, F, G, etc.;
- type of the train (attribute TT) – only moves through the station (1) or must stop at the platform (2);
- current status of track segments (attributes JT1, JT2, JT3, JT4, JOA, JOP) – a segment is free (0) or it is taken (1);
- already prepared routes (attributes B1, B2, B3, etc.) – a route is already set (1) or not (0).

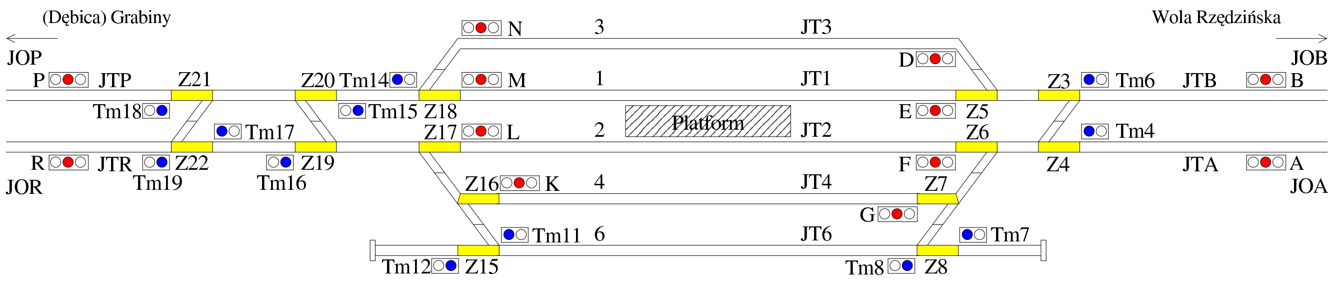


Fig. 1. Czarna Tarnowska – topology of the train station

The decision attributes In and Out represent the input and output routes (that will be prepared for the train) respectively. Domains for these attributes are defined as follows:

$D_{\mathrm{JT}} = \{\mathrm{b}, \mathrm{f}, \mathrm{g}, \mathrm{k}, \mathrm{l}, \mathrm{m}, \mathrm{n}, \mathrm{r}\}$,
$D_{\mathrm{TT}} = \{1, 2\}$,
$D_{\mathrm{JT1}} = D_{\mathrm{JT2}} = D_{\mathrm{JT3}} = D_{\mathrm{JT4}} = D_{\mathrm{JOA}} = D_{\mathrm{JOP}} = \{0, 1\}$,
$D_{\mathrm{B1}} = D_{\mathrm{B2}} = \ldots = D_{\mathrm{N1D}} = \{0, 1\}$,
$D_{\mathrm{In}} = \{\mathrm{b1}, \mathrm{b2}, \mathrm{b3}, \mathrm{b4}, \mathrm{r2}, \mathrm{r4}, \mathrm{none}\}$,
$D_{\mathrm{Out}} = \{\mathrm{f2w}, \mathrm{g2w}, \mathrm{k1d}, \mathrm{l1d}, \mathrm{m1d}, \mathrm{n1d}, \mathrm{none}\}$.

The decision table is presented in table III. Each row represents one decision rule. Let us consider the first rule. It can be represented as follows:

$$\begin{aligned}&(\mathrm{JT} = b) \wedge (\mathrm{TT} = 1) \wedge (\mathrm{JT1} = 0) \wedge (\mathrm{JOP} = 0)\\ &\wedge (\mathrm{B1} = 0) \wedge (\mathrm{B2} = 0) \wedge (\mathrm{B3} = 0) \wedge (\mathrm{B4} = 0)\\ &\wedge (\mathrm{K1D} = 0) \wedge (\mathrm{L1D} = 0) \wedge (\mathrm{M1D} = 0)\\ &\wedge (\mathrm{N1D} = 0) \Rightarrow (\mathrm{b1}, \mathrm{m1d})\end{aligned} \tag{1}$$

It means that if:

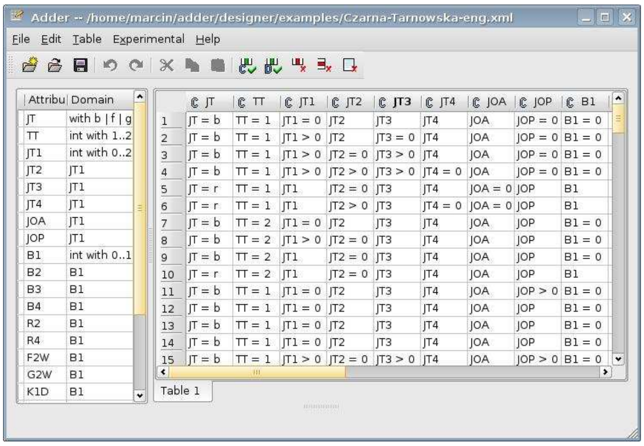
- a train of type 1 is approaching the light signals B (or stands before it), and
- track segments JT1 and JOP are empty, and
- routes B1, B2, B3, B4, K1D, L1D, M1D and N1D are not set,

then routes b1 and m1d should be prepared for the train. It is also worth mentioning that in the considered state other decision are also possible. For example, the following pairs of routes can be used: (b2, l1d), (b3, n1d) or (b4, k1d).

If a row contains empty cells, it means that the values of some attributes are not important for the rule. For example, when rule R1 is considered, it makes no difference what the value of attribute JT2 is.

### A. *Adder DT Designer*

The presented rule-based system has been used in an RTCP-net (Petri net) model of the railway traffic management system. A description of the model can be found in [10]. The decision table has been designed using *Adder DT Designer* ([9]). The tool supports design and analysis of generalised decision tables. It supports three types of attributes' domains: integer, boolean and enumerated data type. The verification stage is included into the design process. At any time, during the design stage, users can check whether a decision table is complete, consistent (deterministic) or if it contains some dependent rules.


Adder -- /home/marcin/adder/designer/examples/Czarna-Tarnowska-eng.xml
File
Edit
Table
Experimental
Help
Table 1

| Attribu | Domain |
|---|---|
| JT | with b \| f \| g |
| TT | int with 1..2 |
| JT1 | int with 0..2 |
| JT2 | JT1 |
| JT3 | JT1 |
| JT4 | JT1 |
| JOA | JT1 |
| JOP | JT1 |
| B1 | int with 0..1 |
| B2 | B1 |
| B3 | B1 |
| B4 | B1 |
| R2 | B1 |
| R4 | B1 |
| F2W | B1 |
| G2W | B1 |
| K1D | B1 |

|  | ℂ JT | ℂ TT | ℂ JT1 | ℂ JT2 | ℂ JT3 | ℂ JT4 | ℂ JOA | ℂ JOP | ℂ B1 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | JT = b | TT = 1 | JT1 = 0 | JT2 | JT3 | JT4 | JOA | JOP = 0 | B1 = 0 |
| 2 | JT = b | TT = 1 | JT1 > 0 | JT2 | JT3 = 0 | JT4 | JOA | JOP = 0 | B1 = 0 |
| 3 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | JT4 | JOA | JOP = 0 | B1 = 0 |
| 4 | JT = b | TT = 1 | JT1 > 0 | JT2 > 0 | JT3 > 0 | JT4 = 0 | JOA | JOP = 0 | B1 = 0 |
| 5 | JT = r | TT = 1 | JT1 | JT2 = 0 | JT3 | JT4 | JOA = 0 | JOP | B1 |
| 6 | JT = r | TT = 1 | JT1 | JT2 > 0 | JT3 | JT4 = 0 | JOA = 0 | JOP | B1 |
| 7 | JT = b | TT = 2 | JT1 = 0 | JT2 | JT3 | JT4 | JOA | JOP | B1 = 0 |
| 8 | JT = b | TT = 2 | JT1 > 0 | JT2 = 0 | JT3 | JT4 | JOA | JOP | B1 = 0 |
| 9 | JT = b | TT = 2 | JT1 | JT2 = 0 | JT3 | JT4 | JOA | JOP | B1 = 0 |
| 10 | JT = r | TT = 2 | JT1 | JT2 = 0 | JT3 | JT4 | JOA | JOP | B1 |
| 11 | JT = b | TT = 1 | JT1 = 0 | JT2 | JT3 | JT4 | JOA | JOP > 0 | B1 = 0 |
| 12 | JT = b | TT = 1 | JT1 = 0 | JT2 | JT3 | JT4 | JOA | JOP | B1 = 0 |
| 13 | JT = b | TT = 1 | JT1 = 0 | JT2 | JT3 | JT4 | JOA | JOP | B1 = 0 |
| 14 | JT = b | TT = 1 | JT1 = 0 | JT2 | JT3 | JT4 | JOA | JOP | B1 = 0 |
| 15 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | JT4 | JOA | JOP > 0 | B1 = 0 |

Fig. 2. Adder DT Designer

An example of *Adder DT Designer* session is shown in Fig. 2. The tool is a free software covered by the GNU Library General Public License. It is being implemented in the GNU/Linux environment by the use of the Qt Open Source Edition. More information about *Adder DT Designer* and the current version of the tool can be found at *http://fm.ia.agh.edu.pl*.

## IV. Exclusion rule-based system

To apply the rule R1 considered in the previous section, it is necessary to check twelve conditions. Further to that, to apply any of the rules presented in Table II, at least seven conditions must be checked. Let us consider Table II that presents mutually exclusive routes. If route B1 is already set, then it cannot be set once again at the same time and also none of routes B2, B3, B4 and N1D can be set.

Suppose the current state is defined as follows: JT $= b$, TT $= 1$, $B1 = 1$ and all other attributes are equal to zero. Let $\mathcal{D}_0$ denote the set of all possible decisions:

$$\begin{aligned}\mathcal{D}_0 = \{&(\mathrm{b1}, \mathrm{m1d}), (\mathrm{b2}, \mathrm{l1d}), (\mathrm{b3}, \mathrm{n1d}), (\mathrm{b4}, \mathrm{k1d}),\\ &(\mathrm{r2}, \mathrm{f2w}), (\mathrm{r4}, \mathrm{g2w}), (\mathrm{b1}, \mathrm{none}), (\mathrm{b2}, \mathrm{none}),\\ &(\mathrm{b3}, \mathrm{none}), (\mathrm{b4}, \mathrm{none}), (\mathrm{r2}, \mathrm{none}), (\mathrm{r4}, \mathrm{none}),\\ &(\mathrm{none}, \mathrm{k1d}), (\mathrm{none}, \mathrm{l1d}), (\mathrm{none}, \mathrm{m1d}),\\ &(\mathrm{none}, \mathrm{n1d}), (\mathrm{none}, \mathrm{f2w}), (\mathrm{none}, \mathrm{g2w}), \}\end{aligned} \tag{2}$$

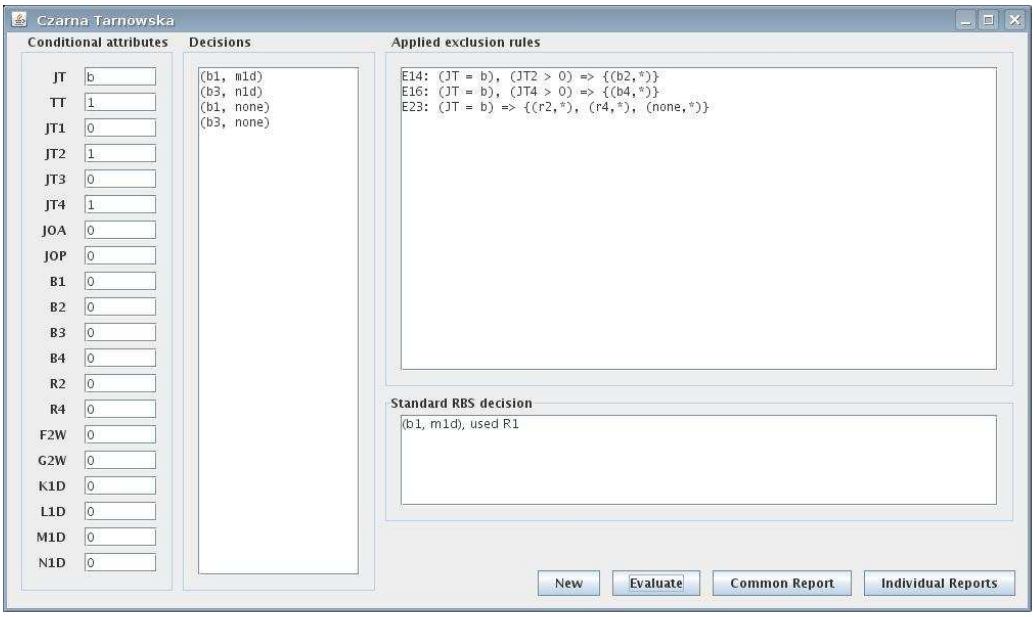
If attribute $B1$ is equal to 1, then the following decisions are impermissible in the considered state: $(\mathrm{b1}, \mathrm{m1d})$, $(\mathrm{b2}, \mathrm{l1d})$,

TABLE III
DECISION TABLE

| | JT | TT | JT1 | JT2 | JT3 | JT4 | JOA | JOP | B1 | B2 | B3 | B4 | R2 | R4 | F2W | G2W | K1D | L1D | M1D | N1D | In | Out |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R1 | JT = b | TT = 1 | JT1 = 0 | | | | | JOP = 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | b1 | m1d |
| R2 | JT = b | TT = 1 | JT1 > 0 | | JT3 = 0 | | | JOP = 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | b3 | n1d |
| R3 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | | | JOP = 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | b2 | l1d |
| R4 | JT = b | TT = 1 | JT1 > 0 | JT2 > 0 | JT3 > 0 | JT4 = 0 | | JOP = 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | b4 | k1d |
| R5 | JT = r | TT = 1 | | JT2 = 0 | | | JOA = 0 | | | B2 = 0 | | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | | | r2 | f2w |
| R6 | JT = r | TT = 1 | | JT2 > 0 | | JT4 = 0 | JOA = 0 | | | B2 = 0 | | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | | | r4 | g2w |
| R7 | JT = b | TT = 2 | JT1 = 0 | | | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | | | | N1D = 0 | b1 | none |
| R8 | JT = b | TT = 2 | JT1 > 0 | JT2 = 0 | | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | | | | | b2 | none |
| R9 | JT = b | TT = 2 | | JT2 = 0 | | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 1 | b2 | none |
| R10 | JT = r | TT = 2 | | JT2 = 0 | | | | | | B2 = 0 | | B4 = 0 | R2 = 0 | R4 = 0 | | G2W = 0 | K1D = 0 | L1D = 0 | | | r2 | none |
| R11 | JT = b | TT = 1 | JT1 = 0 | | | | | JOP > 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | | | | N1D = 0 | b1 | none |
| R12 | JT = b | TT = 1 | JT1 = 0 | | | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 1 | L1D = 0 | M1D = 0 | N1D = 0 | b1 | none |
| R13 | JT = b | TT = 1 | JT1 = 0 | | | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 1 | M1D = 0 | N1D = 0 | b1 | none |
| R14 | JT = b | TT = 1 | JT1 = 0 | | | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 0 | M1D = 1 | N1D = 0 | b1 | none |
| R15 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | | | JOP > 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | | | | | b2 | none |
| R16 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 1 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | | | b2 | none |
| R17 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | K1D = 1 | L1D = 0 | M1D = 0 | N1D = 0 | b2 | none |
| R18 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 1 | M1D = 0 | N1D = 0 | b2 | none |
| R19 | JT = b | TT = 1 | JT1 > 0 | JT2 = 0 | JT3 > 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 1 | N1D = 0 | b2 | none |
| R20 | JT = b | TT = 1 | | JT2 = 0 | JT3 > 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 1 | b2 | none |
| R21 | JT = b | TT = 1 | JT1 > 0 | | JT3 = 0 | | | JOP > 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | | | | | b3 | none |
| R22 | JT = b | TT = 1 | JT1 > 0 | | JT3 = 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 1 | L1D = 0 | M1D = 0 | N1D = 0 | b3 | none |
| R23 | JT = b | TT = 1 | JT1 > 0 | | JT3 = 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 1 | M1D = 0 | N1D = 0 | b3 | none |
| R24 | JT = b | TT = 1 | JT1 > 0 | | JT3 = 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 0 | M1D = 1 | N1D = 0 | b3 | none |
| R25 | JT = b | TT = 1 | | | JT3 = 0 | | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | | | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 1 | b3 | none |
| R26 | JT = b | TT = 1 | JT1 > 0 | JT2 > 0 | JT3 > 0 | JT4 = 0 | | JOP > 0 | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | | | | | b4 | none |
| R27 | JT = b | TT = 1 | JT1 > 0 | JT2 > 0 | JT3 > 0 | JT4 = 0 | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 1 | L1D = 0 | M1D = 0 | N1D = 0 | b4 | none |
| R28 | JT = b | TT = 1 | JT1 > 0 | JT2 > 0 | JT3 > 0 | JT4 = 0 | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 1 | M1D = 0 | N1D = 0 | b4 | none |
| R29 | JT = b | TT = 1 | JT1 > 0 | JT2 > 0 | JT3 > 0 | JT4 = 0 | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 1 | N1D = 0 | b4 | none |
| R30 | JT = b | TT = 1 | | JT2 > 0 | JT3 > 0 | JT4 = 0 | | | B1 = 0 | B2 = 0 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 0 | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 1 | b4 | none |
| R31 | JT = r | TT = 1 | | JT2 = 0 | | | JOA > 0 | | | B2 = 0 | | B4 = 0 | R2 = 0 | R4 = 0 | | G2W = 0 | K1D = 0 | L1D = 0 | | | r2 | none |
| R32 | JT = r | TT = 1 | | JT2 = 0 | | | | | | B2 = 0 | | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 1 | G2W = 0 | K1D = 0 | L1D = 0 | | | r2 | none |
| R33 | JT = r | TT = 1 | | JT2 > 0 | | JT4 = 0 | JOA > 0 | | | | | B4 = 0 | R2 = 0 | R4 = 0 | | | K1D = 0 | L1D = 0 | | | r4 | none |
| R34 | JT = r | TT = 1 | | | | JT4 = 0 | | | B1 = 0 | B2 = 1 | B3 = 0 | B4 = 0 | R2 = 0 | R4 = 0 | | | K1D = 0 | L1D = 0 | | | r4 | none |
| R35 | JT = r | TT = 1 | | JT2 > 0 | | JT4 = 0 | | | | | | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 1 | G2W = 0 | K1D = 0 | L1D = 0 | | | r4 | none |
| R36 | JT = r | TT = 1 | | | | JT4 = 0 | | | | | | B4 = 0 | R2 = 0 | R4 = 0 | F2W = 0 | G2W = 1 | K1D = 0 | L1D = 0 | | | r4 | none |
| R37 | JT = k | TT = 1 | | | | JT4 > 0 | | JOP = 0 | | | | | R2 = 0 | R4 = 0 | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | none | k1d |
| R38 | JT = l | | | JT2 > 0 | | | | JOP = 0 | | | | | R2 = 0 | R4 = 0 | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | none | l1d |
| R39 | JT = m | | JT1 > 0 | | | | | JOP = 0 | | | | | | | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | none | m1d |
| R40 | JT = n | TT = 1 | | | JT3 > 0 | | | JOP = 0 | B1 = 0 | | | | | | | | K1D = 0 | L1D = 0 | M1D = 0 | N1D = 0 | none | n1d |
| R41 | JT = f | | | JT2 > 0 | | | JOA = 0 | | | B2 = 0 | | B4 = 0 | | | F2W = 0 | G2W = 0 | | | | | none | f2w |
| R42 | JT = g | TT = 1 | | | | JT4 > 0 | JOA = 0 | | | B2 = 0 | | B4 = 0 | R2 = 0 | | F2W = 0 | G2W = 0 | | | | | none | g2w |

$E_1\colon (\mathrm{B1} = 1) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (*, \mathrm{n1d})\}$<br>
$E_2\colon (\mathrm{B2} = 1) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w})\}$<br>
$E_3\colon (\mathrm{B3} = 1) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *)\}$<br>
$E_4\colon (\mathrm{B4} = 1) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w})\}$<br>
$E_5\colon (\mathrm{R2} = 1) \Rightarrow \{(\mathrm{b2}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{g2w}), (*, \mathrm{k1d}), (*, \mathrm{l1d})\}$<br>
$E_6\colon (\mathrm{R4} = 1) \Rightarrow \{(\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{k1d}), (*, \mathrm{l1d})\}$<br>
$E_7\colon (\mathrm{F2W} = 1) \Rightarrow \{(\mathrm{b2}, *), (\mathrm{b4}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w})\}$<br>
$E_8\colon (\mathrm{G2W} = 1) \Rightarrow \{(\mathrm{b2}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w})\}$<br>
$E_9\colon (\mathrm{K1D} = 1) \Rightarrow \{(\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{10}\colon (\mathrm{L1D} = 1) \Rightarrow \{(\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{11}\colon (\mathrm{M1D} = 1) \Rightarrow \{(*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{12}\colon (\mathrm{N1D} = 1) \Rightarrow \{(\mathrm{b1}, *), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{13}\colon (\mathrm{JT} = b) \wedge (\mathrm{JT1} > 0) \Rightarrow \{(\mathrm{b1}, *)\}$<br>
$E_{14}\colon (\mathrm{JT} = b) \wedge (\mathrm{JT2} > 0) \Rightarrow \{(\mathrm{b2}, *)\}$<br>
$E_{15}\colon (\mathrm{JT} = b) \wedge (\mathrm{JT3} > 0) \Rightarrow \{(\mathrm{b3}, *)\}$<br>
$E_{16}\colon (\mathrm{JT} = b) \wedge (\mathrm{JT4} > 0) \Rightarrow \{(\mathrm{b4}, *)\}$<br>
$E_{17}\colon (\mathrm{JT} = r) \wedge (\mathrm{JT2} > 0) \Rightarrow \{(\mathrm{r2}, *)\}$<br>
$E_{18}\colon (\mathrm{JT} = r) \wedge (\mathrm{JT4} > 0) \Rightarrow \{(\mathrm{r4}, *)\}$<br>
$E_{19}\colon (\mathrm{JT} = b) \wedge (\mathrm{TT} = 2) \Rightarrow \{(\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{b1}, \mathrm{m1d}), (\mathrm{b2}, \mathrm{l1d})\}$<br>
$E_{20}\colon (\mathrm{JT} = r) \wedge (\mathrm{TT} = 2) \Rightarrow \{(\mathrm{r4}, *), (\mathrm{r2}, \mathrm{f2w})\}$<br>
$E_{21}\colon (\mathrm{JOA} > 0) \Rightarrow \{(*, \mathrm{f2w}), (*, \mathrm{g2w})\}$<br>
$E_{22}\colon (\mathrm{JOP} > 0) \Rightarrow \{(*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{23}\colon (\mathrm{JT} = b) \Rightarrow \{(\mathrm{r2}, *), (\mathrm{r4}, *), (\mathrm{none}, *)\}$<br>
$E_{24}\colon (\mathrm{JT} = r) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{none}, *)\}$<br>
$E_{25}\colon (\mathrm{JT} = k) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{26}\colon (\mathrm{JT} = l) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w}), (*, \mathrm{k1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{27}\colon (\mathrm{JT} = m) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w}), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{n1d})\}$<br>
$E_{28}\colon (\mathrm{JT} = n) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{f2w}), (*, \mathrm{g2w}), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d})\}$<br>
$E_{29}\colon (\mathrm{JT} = f) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{g2w}), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$<br>
$E_{30}\colon (\mathrm{JT} = g) \Rightarrow \{(\mathrm{b1}, *), (\mathrm{b2}, *), (\mathrm{b3}, *), (\mathrm{b4}, *), (\mathrm{r2}, *), (\mathrm{r4}, *), (*, \mathrm{f2w}), (*, \mathrm{k1d}), (*, \mathrm{l1d}), (*, \mathrm{m1d}), (*, \mathrm{n1d})\}$

Fig. 3. Exclusion rule-based system



Conditional attributes

| | |
|---|---|
| JT | b |
| TT | 1 |
| JT1 | 0 |
| JT2 | 1 |
| JT3 | 0 |
| JT4 | 1 |
| JOA | 0 |
| JOP | 0 |
| B1 | 0 |
| B2 | 0 |
| B3 | 0 |
| B4 | 0 |
| R2 | 0 |
| R4 | 0 |
| F2W | 0 |
| G2W | 0 |
| K1D | 0 |
| L1D | 0 |
| M1D | 0 |
| N1D | 0 |

Decisions

```
(b1, m1d)
(b3, n1d)
(b1, none)
(b3, none)
```

Applied exclusion rules

```
E14: (JT = b), (JT2 > 0) => {(b2,*)}
E16: (JT = b), (JT4 > 0) => {(b4,*)}
E23: (JT = b) => {(r2,*), (r4,*), (none,*)}
```

Standard RBS decision

```
(b1, m1d), used R1
```

Fig. 4. Comparison of rule-based systems

$(\mathrm{b3},\mathrm{n1d})$, $(\mathrm{b4},\mathrm{k1d})$, $(\mathrm{b1},\mathrm{none})$, $(\mathrm{b2},\mathrm{none})$, $(\mathrm{b3},\mathrm{none})$, $(\mathrm{b4},\mathrm{none})$, $(\mathrm{none},\mathrm{n1d})$.

For simplicity, the asterisk $(*)$ will be used to denote any value the corresponding entry can take. Thus, the first exclusion rule $E1$ takes the following form:

$$E_1\colon (\mathrm{B1}=1) \Rightarrow \{(\mathrm{b1},*),(\mathrm{b2},*),(\mathrm{b3},*),(\mathrm{b4},*),(*,\mathrm{n1d})\} \tag{3}$$

Each row of Table II generates an exclusion decision rule. Thus, the constructed exclusion rule-based system will also contain next eleven rules (see Fig. 3, rules from $E_2$ to $E_{12}$).

Let us focus on the train station topology (see Fig. 1). If a train position (attribute TJ) is equal to $b$ and track segment JT1 is taken by another train, then any route leading through the track segment cannot be taken under consideration. Thus, we have the following exclusion rule:

$$E_{13}\colon (\mathrm{JT}=b) \wedge (\mathrm{JT1}>0) \Rightarrow \{(\mathrm{b1},*)\} \tag{4}$$

Similarly, we can investigate situations when one of track segments JT2, JT3, JT4 is busy, or a train position is equal to $r$ and track segment JT2 or JT4 is busy – see rules from $E_{14}$ to $E_{19}$.

If a train type (attribute TT) is equal to 2, then it should stop at the platform. In such a case only an input route must be prepared and routes leading through track segments JT3 and JT4 cannot be taken under consideration – see rules $E_{20}$ and $E_{21}$. Moreover, an output route can be prepared only if the corresponding output track segment is free (track segments JOA and JOP) – see rules $E_{22}$ and $E_{23}$.

The last group of exclusion rules corresponds to the direction of trains moving. If a train position is equal to $b$, then the train moves from 'east' to 'west' and first of all an input route for the train must be prepared. Thus, any decisions with the input route equal to *none* or with routes leading to east are impermissible in such a situation. Therefore, the ExRBS must contain the following exclusion rule:

$$E_{23}\colon (\mathrm{JT}=b) \Rightarrow \{(\mathrm{r2},*),(\mathrm{r4},*),(\mathrm{none},*)\} \tag{5}$$

In a similar way, the other seven train positions can be investigated. Finally, the ExRBS must be extended with seven exclusion rules (see Fig. 3, rules from $E_{23}$ to $E_{30}$).

Executing a single exclusion rule removes some impermissible decisions from the set of all possible ones. The set of decision excluded by an exclusion rule will be denoted by $\mathcal{E}(E_i)$. Hence, for example, the set $\mathcal{E}(E_1)$ contains the following elements:

$$\begin{aligned}\mathcal{E}(E_1) = \{&(\mathrm{b1},\mathrm{m1d}),(\mathrm{b2},\mathrm{l1d}),(\mathrm{b3},\mathrm{n1d}),\\ &(\mathrm{b4},\mathrm{k1d}),(\mathrm{b1},\mathrm{none}),(\mathrm{b2},\mathrm{none}),\\ &(\mathrm{b3},\mathrm{none}),(\mathrm{b4},\mathrm{none}),(\mathrm{none},\mathrm{n1d})\}\end{aligned} \tag{6}$$

Let $\mathcal{D}_i$ denote the set of permissible decisions after executing the rule $E_i$. The following equality holds:

$$\mathcal{D}_{i+1} = \mathcal{D}_i - \mathcal{E}(E_{i+1}) \tag{7}$$

After executing all exclusion rules with satisfied preconditions, the set contains all decisions permissible in the considered state.

For the state considered in this section, rules $E1$ and $E23$ can be applied. The final set of permissible decisions $\mathcal{D}$ is empty. It means that at the moment no route for the train can be prepared. Another example—a state and decisions generated by the decision table and the exclusion rule-based system—is presented in Fig. 4.

The decision table presented in the previous section is complete and deterministic. It means that if it is possible to generate a decision for any acceptable state, then only one decision is generated. To compare the two presented rule-based systems, a Java software has been implemented and decisions generated by the systems for any acceptable state have been compared.[1] For any state the ExRBS generates at least the same decision as the decision table (for many states ExRBS generates more than one possibility).

## V. Conclusion

The idea of exclusion rule-based systems has been given in this paper, and a comparison has been made between decision tables and exclusion rule-based systems. The latter ones can be treated as an alternative method of designing rule-based systems. The presented approach has been illustrated by the use of an example of a railway traffic management system. In this case, ExRBS is significantly more readable and adaptable. In contrast to decision tables, an ExRBS for a train station with different topology and size can be easily constructed in a very similar way. Moreover, the exclusion rule-based system contains less decision rules then the corresponding decision table and the design of the ExRBS was significantly less time-consuming.

## References

[1] P. Jackson, *Introduction to Expert Systems*. Addison-Wesley, 1999.

[2] J. Liebowitz, *The Handbook of Applied Expert Systems*. CRC Press, 1998.

[3] A. Lig˛eza, *Logical Foundations of Rule-Based Systems*. Berlin, Heidelberg: Springer-Verlag, 2006.

[4] M. Negnevitsky, *Artificial Intelligence. A Guide to Intelligent Systems*. Harlow, England; London; New York: Addison-Wesley, 2002.

[5] G. J. Nalepa and A. Ligęza, "Designing reliable web security systems using rule-based systems approach," in *Advances in Web Intelligence: first international Atlantic Web Intelligence Conference AWIC 2003*, ser. Lecture Notes in Artificial Intelligence, E. Menasalvas, J. Segovia, and P. S. Szczepaniak, Eds., vol. 2663. Springer-Verlag, 2003, pp. 124–133.

[6] B. Fryc, K. Pancerz, and Z. Suraj, "Approximate Petri nets for rule-based decision making," in *Proceedings of the 4th International Conference on Rough Sets and Current Trends in Computing, RSCTC 2004*, ser. Lecture Notes in Artificial Intelligence, J. Komorowski and S. Tsumoto, Eds., vol. 3066. Springer-Verlag, 2004, pp. 733–742.

[7] A. Davis, "A comparison of techniques for the specification of external system bahavior," *Communication of the ACM*, vol. 31, no. 9, pp. 1098–1115, 1988.

[8] A. Ligęza, "Toward logical analysis of tabular rule-based systems," *International Journal of Intelligent Systems*, vol. 16, no. 3, pp. 333–360, 2001.

[9] M. Szpyrka, "Design and analysis of rule-based systems with Adder Designer," in *Knowledge-Driven Computing*, ser. Studies in Computational Intelligence, C. Cotta, S. Reich, R. Schaefer, and A. Ligęza, Eds. Springer-Verlag, 2008, vol. 102, pp. 255–271.

[10] ——, "Modelling and analysis of real-time systems with RTCP-nets," in *Petri Net, Theory and Applications*, V. Kordic, Ed. Vienna, Austria: I-Tech Education and Publishing, 2008, ch. 2, pp. 17–40.

[1] For more details see `http://fm.ia.agh.edu.pl`

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 243–250

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Ontological Matchmaking in a *Duty Trip Support* Application in a Virtual Organization

Michal Szymczak,
Grzegorz Frackowiak,
Maria Ganzha and
Marcin Paprzycki
System Research Institute,
Polish Academy of Sciences,
Poland
Email: maria.ganzha@ibspan.waw.pl

Sang Keun Rhee,
Jihye Lee,
Young Tae Sohn,
Jae Kwan Kim
Yo-Sub Han and
Myon-Woong Park
Korea Institute of Science and Technology,
Seoul, Korea

***Abstract*—In our work, an agent-based system supporting workers in fulfilling their roles in a virtual organization has as its centerpiece ontologically demarcated data. In such system, ontological matchmaking is one of key functionalities in provisioning of personalized information. Here, we discuss how matchmaking will be facilitated in a *Duty Trip Support* application. Due to the nature of the application, particular attention is paid to the geospatial data processing.**

## I. Introduction

CURRENTLY, we are developing an agent-based system supporting resource management in a virtual organization. While it is often assumed that the notion of virtual organization should be applied when workers are geographically distributed ([1]), we do not make this assumption. Instead, we consider as "virtualization" process in which a real organization is "mapped" into a virtual one, where virtual can be understood as "existing electronically" (see, also [2]). In such virtual organization (1) its structure is represented by software agents and their interactions, while (2) the organization itself and its domain of operation are ontologically described. Here, we recognize need for (i) an ontology of an organization (e.g. specifying who has access to which resource, or which department does a given person work for), and (ii) a domain specific ontology (e.g. ontology of a car repair shop, specifying areas of expertise and skills of individual workers); see [3], [4], [5], [6] for summary of results obtained thus far.

One of main reasons for utilizing ontologies is that they allow for application of semantic reasoning. The aim of this paper is to describe how a specific type of such reasoning—ontological matching—can be used in the context of an applications currently under development. While in [7] we have considered matchmaking involved in the *Grant Announcement Support*, here we consider a *Duty Trip Support*. One of the key concepts that we will explore is geospatial matchmaking. To this effect, in the next section, we introduce the *Duty Trip Support* (*DTS*) application, following with an introduction to matchmaking processes that take place in the system. Next, we discuss specific matchmaking taking place in the *DTS*. In this section we also present in detail our ontological matchmaking algorithm.

## II. *Duty Trip Support* application

In our earlier work [5] we have introduced a scientist, Mr. Jackie Chan, employed in a Science Institute in Aberdeen, Hong Kong, China, who goes on a duty trip to Finland and will utilize a *Duty Trip Support* (*DTS*) application. It is expected that the *DTS* will be able to suggest to travelers places to stay and eat (based on their personal preferences and experiences of other travelers, e.g. from the same institution). This is an example of personalized information delivery that takes into account cultural and dietary differences between, for instance, Japan and Germany. A complete description of proposed functionalities and processes involved in the *DTS* application (including a complete UML sequence diagram) can be found in [8], [4], [5]. Here we focus our attention on ontological matchmaking utilized in the *DTS*. Note that majority of examples listed here (ontology classes and properties, in particular) are based on our previous work (see, [4], [5] for more details). Doing so, allows us to shed more light on these examples and matching processes taking place in the system, which were only outlined before, while preserving continuity and building a complete picture of the process.

Let us now reintroduce ontology class instance samples and start from a listing of *City* and *Country* instances which include geospatial information. Two countries (China and Finland) and three cities (Aberdeen, Oulu and Rovaniemi) are listed (each city located in one of the two countries).

```
geo:FinlandCountry a onto:Country;
        onto:name "Finland"^^xsd:string.
geo:ChinaCountry a onto:Country;
        onto:name "China"^^xsd:string.
geo:OuluCity a onto:City;
        onto:name "Oulu"^^xsd:string;
        onto:long "25,467"^^xsd:float;
        onto:lat "65,017"^^xsd:float;
        onto:isInCountry :FinlandCountry.
geo:RovaniemiCity a onto:City;
        onto:name "Rovaniemi"^^xsd:string;
        onto:long "25,8"^^xsd:float;
        onto:lat "66,567"^^xsd:float;
```

978-83-60810-14-9/08/$25.00 © 2008 IEEE

```
        onto:isInCountry  :FinlandCountry.
geo:AberdeenCity a onto:City;
        onto:name "Aberdeen"^^xsd:string;
        onto:long "114,15"^^xsd:float;
        onto:lat "22,25"^^xsd:float;
        onto:isInCountry  :ChinaCountry.
```

Next, let us recall the sample employee (Mr. Chan). As discussed in [4], [5], each employee is associated with several profiles. Note, however, that some resources introduced in this document have only a single profile assigned. In general, resource detailed information is stored in its profile in order to facilitate adaptability in the system, which assumes that profile extensibility, robustness and resource access control are required. Therefore, saving resource attributes within profiles allows to easily extend and adapt individual resource records (see, also [9]). Below we present a snippet based on the *Employee Profile*, which consists of a *Personal Profile* and an *Experience Profile*:

```
:Employee\#1 a onto:ISTPerson;
    onto:id "1234567890"^^xsd:string;
    onto:hasProfile (:Employee\#1PProfile,
                     :Employee\#1EProfile),
    onto:belongsToOUs (:GOU).
:ResearchOU a onto:OrganizationUnit;
    onto:name ‘‘Researchers Organization
                        Unit’’^^xsd:string.
```

In this example the *Employee#1PProfile*—the *Personal Profile*, presented next—describes the “human resource (HR) properties” of an employee. In what follows we use only basic properties: *fullname*, *gender* and *birthday*; as well as the *belongsToOUs* property, which indicates Mr. Chan’s position in the organization (the *Organizational Unit* he works for). Let us stress that a complete list of HR properties is organization-dependent and is instantiated within the ontology of a given organization.

```
:Employee\#1PProfile a onto:ISTPersonalProfile;
  onto:belongsTo :Employee\#1;
  person:fullname ‘‘Yao Chan’’^^xsd:string;
  person:gender person:Male;
  person:birthday
        ‘‘1982-01-01T00:00:00’’^^xsd:dateTime.
```

The second profile of *Employee#1* (Mr. Chan) that we have introduced, is the *Experience Profile* that demarcates his specialization in terms of fields of knowledge and project experience. Here, codes for the fields of knowledge specification originate from the KOSEF (Korea Science and Engineering Foundation) [10]. Obviously, *any* classification of fields of knowledge/expertise could be applied here (appropriately represented within the ontology of an organization).

```
:Employee\#1EProfile a onto:ISTExperienceProfile;
  onto:belongsTo :Employee\#1;
  onto:doesResearchInFields
    scienceNamespace:Volcanology-13105,
    scienceNamespace:Paleontology-13108,
    scienceNamespace:Geochronology-13204;
  onto:knowsFields
    [a onto:Knowledge;
    onto:knowledgeObject
        scienceNamespace:Volcanology-13105;
    onto:knowledgeLevel "0.25"^^xsd:float],
    [a onto:Knowledge;
    onto:knowledgeObject
        scienceNamespace:Paleontology-13108;
    onto:knowledgeLevel "0.15"^^xsd:float],
    [a onto:Knowledge;
    onto:knowledgeObject
        scienceNamespace:Geochronology-13204;
    onto:knowledgeLevel "0.90"^^xsd:float];
  onto:managesProjects (:Project1).
```

According to the *ISTExperience profile*, Mr. Chan specializes in *Volcanology*, *Paleontology* and *Geochronology*. Level of knowledge in each of these areas is expressed as a sample real value; respectively: 0.25, 0.15, 0.9. Here, we assume that values describing the level of knowledge in specific fields are a result of self-assessment of an employee. However, in [11] we have proposed mechanisms for human-resource adaptability, which can be utilized to automatically (or semi-automatically) adapt level of knowledge on the basis of, for instance, work and training history of the employee. Furthermore, note that professional test (existing in most fields) can also be directly used as a method for knowledge level assessment. Now, *Employee#1* who is described with that profile manages project *Project1*. It is a scientific project in *Volcanology* (see below).

```
:Project1 a onto:ISTProject;
 onto:managedBy :Employee\#1;
 onto:period
 [a onto:Period;
 onto:from "2008-06-01T00:00:00"^^xsd:dateTime;
 onto:to "2009-05-31T00:00:00"^^xsd:dateTime];
 onto:fieldsRef scienceNamespace:Volcanology-13105;
 onto:projectTitle ‘‘Very Important Volcanology
               Scientific Project’’^^xsd:string.
```

To be able to illustrate matching processes taking place within the *Duty Trip Support* application, we will now introduce instances of a *Contact Person* (*:ContactPerson#1*) and a *Duty Trip Report* (*:DTR#1*); see [5], [7] for additional details).

```
:ContactPerson\#1 a onto:ContactPerson;
  onto:hasProfile :ContactPersonProfile\#1.
:ContactPersonProfile\#1
                a onto:ContactPersonProfile;
  person:fullname
           ‘‘Mikka Korteleinen’’^^xsd:string;
  person:gender person:Male;
  person:birthday
        ‘‘1967-11-21T00:00:00’’^^xsd:dateTime;
  onto:doesResearch science:Paleontology-13108,
                    science:Volcanology-13105;
  onto:locatedAt geo:RovaniemiCity;
  onto:belongsTo :ContactPerson\#1.
:ContactPerson\#2 a onto:ContactPerson;
  onto:hasProfile :ContactPersonProfile\#2.
:ContactPersonProfile\#2
                a onto:ContactPersonProfile;
  person:fullname
           ‘‘Juno Viini’’^^xsd:string;
  person:gender person:Male;
  person:birthday
        ‘‘1957-01-15T00:00:00’’^^xsd:dateTime;
  onto:doesResearch science:Geochronology-13204;
  onto:locatedAt geo:RovaniemiCity;
  onto:belongsTo :ContactPerson\#2.
:DTR\#1 a onto:ISTDutyTripReport;
  onto:hasProfile (:DTRProfile\#1).
:DTRProfile\#1 a onto:ISTDutyTripReportProfile;
  onto:destination geo:OuluCity;
  onto:traveler :Employee\#1;
  onto:status dtStatusNamespace:Application;
  [a onto:Period;
  onto:from ‘‘2008-06-07T00:00:00’’^^xsd:dateTime;
  onto:to ‘‘2008-06-19T00:00:00’’^^xsd:dateTime.].
  onto:stayedAt hot:OuluRadisonSAS
  onto:expense [a onto:SingleCost;
        ‘‘4000’’^^xsd:float;
onto:expenseCurrency ‘‘USD’’^^xsd:string.]
onto:purpose ‘‘Conference’’^^xsd:string;
onto:belongsTo :DTR\#1.
```

As we will see, *Contact Person*s will be suggested (though for a different reason) by our system as someone who Mr. Chan should visit. First, Mikka Korteleinen, is defined through the *ContactPerson#1* and the *ContactPersonProfile#1* objects. The latter object defines Mr. Korteleinen's field of specialization to be *Paleontology* and *Volcanology*. Here, the level of expertise of Mr. Korteleinen is not specified, because it is assumed that such data is a result of self-assessment of the person, or processes taking place internally in her/his organization (see above); and as such is not available to the *DTS* of Mr. Chan's organization. However, note that the very fact that a potential contact person is in a system is very likely going to be a result of a personal meeting with her/him, followed by an employee introducing the contact information into the system. Such employee, could potentially assess not only area(s) of expertise, but also level of knowledge in each one of them. This possibility will be explored in the future. Now, we can observe that the profile of Mr. Korteleinen informs us that he can be reached in Rovaniemi, Finland. Second, the profile of Mr. Juno Viini was defined. According to this example he can be found in Rovaniemi as well, while his research interest is *Geochronology*. Separately, note that the *Duty Trip Report* is a basic resource associated with any *Duty Trip*. It is defined through the *DTR#1* and the *DTRProfile#1* objects and represents a required set of information associated by the organization with a *Duty Trip*. It describes travel details details, such as:

```
destination: Oulu, Finland,
status: application,
purpose: a conference.
```

Here, we can see that Mr. Chan plans to travel to Oulu, Finland to a conference and he has applied for this *Duty Trip* (in the case his travel is approved, the field *status* will change its value to *approved*). Our aim in this paper is to show how ontological matchmaking can be used to find cities near-by the city where Mr. Chan is to travel to, and person(s) that he may want do consider visiting during his trip.

## III. Matching in the system

### A. General idea

What is needed to achieve our goal is to be able to establish measure of distance (similarity) between two (or more) instances of ontologically demarcated data (here, between researchers considered in the context of cities they reside in). Before proceeding, let us note first that in our work we have made an important simplifying assumption. Across all currently developed applications (which are to work *within* an organization), a single ontology is used. Therefore, we do not have to deal with problems related to *ontology matching/integration* (where an attempt is made to establish "common understanding" between two, or more, ontologies; see, for instance [12], [13], [14]). Since our goal is to establish if specific instances of an ontology are "close enough," our objective should be to define a measure of distance representable as a single number (among others, for ease of comparison). This number can be then compared against a threshold to make a decision if objects are relevant to each-other. Note also that for each application there is a specific *Matching Criteria* that represents the "focus" of the matching process (e.g. research interests, eating preferences, or location). In [7] we have described in general terms process of measuring closeness of objects—*Calculating Relevance*, and presented it in the context of the *Grant Announcement Application*. Here a modified version of that algorithm will be proposed. However, we can use the same point of departure and define the *Matching Criteria* as a tuple (quadruple in this case) $\langle x, q, a, g \rangle$, where:

- $x$ is the selected ontology class instance (source object)
- $q$ is a SPARQL query ([15]) which defines a subset of objects that are considered potentially relevant (this is the above mentioned focus of the matchmaking process) and will be matched against the source object $x$
- $a \geq 0$, specifies threshold of closeness between objects to be judged actually relevant to each-other
- $g$ is a sub-query processed by the GIS subsystem (in general, this parameter can be omitted—if there is no geospatial query involved; or it can be replaced by one or more different criteria; thus the notion that the *Matching Criteria* is a tuple); this part of the system is responsible for finding cities which are located within a specified distance to a specified city; this sub-query is a triple $\langle gr, gc, ga \rangle$, where:
  - $gr$ is an operator which allows to either limit returned number of cities of possible interest (*AMOUNT* condition) or to specify the maximum distance between the $gc$ and the returned cities (*RADIUS* condition)
  - $gc$ is an URI of a city demarcated with properties of the *City* class of the system ontology
  - $ga$ is the parameter of the $gr$ operator ($gr(gc, ga)$); it either specifies the limit of the number of returned cities or the maximum distance between the $gc$ and the returned cities

### B. Relevance graph

Calculation of "distance" between instances of ontology is based on a graph structure that represents the underlaying *Jena Ontology Model*. Specifically, this *Model* is interpreted as a directed graph $G = (V, E)$ such that (here, reflexive relations are ignored):

$$V: \text{set of nodes, representing all instances,}$$

$$E: \text{set of edges, representing all object properties.}$$

Upon creating edges, the value of the annotation property *voPropertyWeight* of each object property becomes the label of the edge, representing the "distance between two nodes" ("importance of the relationship between two nodes") in the ontology. This concept comes from work of S. Rhee and collaborators (see, [16], [17] for more details), where it was shown how connections between nodes in the graph representing concepts in the ontology can be weighted according to the importance of their relationships. For instance, suppose for a given researcher we have two relations `doesResearchInFields` and `worksForProjects`; and the system attempts to recommend an additional contact person based on those two relations. Here, `worksForProjects` can be regarded to be more important than `doesResearchInFields` because the fact that two persons work at a given time for a similar project can be considered more important for recommending them to eachother than the situation when two persons declare that they have same (or similar) research interests. Currently, in our system we do not have an automatic way of assigning weights to edges in an ontology, and thus weights equal to 1 will be initially used as a default (which is also the approach suggested in [16], [17]). However, we assume that different weights can (and will in the future) be used in the graph representing the proposed ontology. It is also worthy stressing that such values allow us to naturally deal with concepts that are not directly connected in a directed graph (see, [6] for more details). For the sake of precision it can be thus stated that the relevance graph $G = (V, E)$ becomes $G = (V, E, W)$,

$V$ : set of nodes, representing all instances,

$E$ : set of edges, representing all object properties.

$W$ : "importance weights" assigned to all edges

Let us now observe that while weight values assigned to each edge represent the distance between two nodes in the *Model*, some instances connected by a certain relation (i.e. having the same "importance weight") may not have the same "importance" on the level of individuals. For example, a person may have knowledge/interest in three different subject areas, while his/her knowledge/interest level in each area may be different (see the *ISTExperience profile* example in Section II).

Therefore, we can distinguish two levels of "scaling" of importance of ontological relationships. The first one is "within the ontology" and involves relationships between concepts (nodes in the relevance graph). The second one is on the "level of instances" and specifies importance of specific properties to an individual. Therefore, when the ontological distance is calculated, first we have to take into account ontological distance between concepts and, second, to scale it according to individual "interests" of resources involved in matching. As an example of such process, delivery of personalized information is depicted in figure 1.

Here, we can see an employee within an organization. The ontology of an organization and the domain ontology provide us with a (weighted) relevance graph ($G = (V, E, W)$). At the same time, an ontological instance—the employee profile—allows us to scale specific relations in the ontology according to the employees' interests. Both the relevance graph and the individual profile, together with resources, closeness to which is to be established (selected according to the *Matching Criteria*), are the input to the matching algorithm. As an output we obtain list of resources that are relevant to the employee.

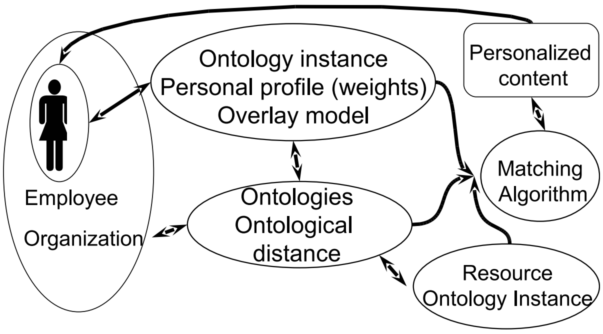


Fig. 1. Top level overview of matchmaking

### C. Geospatial Information Subsystem

Before proceeding to describe in detail the matchmaking algorithm and its utilization in the *Duty Trip Support* application, let us make the following comment on the (optional) GIS sub-query. In general, the first four terms of the *Matching Criteria*, above, allow one to find objects relevant to the given *source object* assuming that there is a (directed) path between these objects in the *Relevance Graph* (these objects are linked with each other). However, one of basic requirements of the system under development is that it will provide geospatial information-based recommendations. Observe that the *Relevance Graph* does not provide natural support for distance calculation in terms of geographical localization (as it is represented in our ontology). Specifically, the *Relevance Graph* is built on the basis of nodes that represent ontology class instances, which in turn are linked by edges that represent properties. In our ontology, geospatial attributes are: *longitude*, *latitude*, *altitude* and *inCountry*. While meaning of the first three is obvious, the latter is a relation between instances of the *Country* and the *City* classes that allows us to model "cities being in a country" property.

SPARQL seems to be most recommended technology for querying the RDF demarcated data and it provides support for computing results of mathematical calculations on *Datatype Properties* values of objects defined in the RDF [18]. The latter functionality is necessary for finding RDF demarcated *City* class instances which meet certain distance criteria in terms of geospatial localization. However, designing full GIS support to be performed by the SPARQL engine would require full information about countries and cities of the world to be stored in the semantic storage. Such an approach would overload the semantic storage and severely influence other SPARQL operations which have to be performed on the RDF demarcated data. On the other hand, distances between nodes of the *Relevance Graph* correspond to (scaled) weights of ontology properties that reflect the semantic distance of certain concepts. This distance is not the geospatial distance. Therefore, in order to avoid semantic storage replication/clustering, which would be necessary if we stored all GIS information in the semantic storage, we decided, for the time being, that the GIS sub-query is going to be processed by a dedicated subsystem that allows us to select only related objects which meet criteria defined in the sub-query. This subsystem returns as a result a list of *City* object URIs which represent cities that meet the geographical localization criteria and, in the case that a particular *City* object does not exist in the semantic storage, it creates the necessary RDF statements. This subsystem is simple and independent and it reduces the volume of the RDF demarcated data stored in the system. Please note that this solution is temporary as we experiment with an alternative that allows to utilize full computational possibilities of SPARQL and keeps RDF data volume as small as possible.

## IV. *Duty Trip Support*-based Matching Example

As noted, one of important functionalities of the *Duty Trip Support* subsystem is to suggest optional activities of an employee who plans a duty trip. In order to give Mr. Chan advice about possible extensions of his duty trip, first, it is necessary to define appropriate *Matching Criteria*. Next, the delivered advice is a result of matching between the *DTR#1* object and instances of the *ContactPerson* class ([19], [7]). Note that in the future such matching will involve also information about food and accommodations (while other features can also be naturally selected). Overall, the matching process in the *DTS* involves the following steps:

1) Construct Matching Criteria $\langle x, q, a, g \rangle$:
   a) $x = DTR\#1$
   b) $q =$

```
PREFIX onto:
<http://rossini.ibspan.waw.pl/
            Ontologies/KIST/KISTVO>
SELECT ?person
WHERE {?person isa onto:ContactPerson.}
 FILTER (onto:locatedAt
                 temp:gisResults-multi).
```

   c) $a = \frac{1}{40}$
   d) $g = [gc, gr, ga]$ :

$$gc = OuluCity,$$
$$gr = RADIUS,$$
$$ga = 200.$$

   The *Criteria* defined above can be stated as: find a potentially interesting (in terms of professional interests) person who resides not further than 200 km away from Oulu.
2) Execute the GIS query $g$. To do this, invoke the GIS interface method which returns references to objects which represent (known to the system) cities located within 200 km distance from the city of Oulu. Results are chosen from all cities for which at least one RDF object in the semantic storage exists. The result in our example is: RovaniemiCity for which the distance from *OuluCity* equals to 173.168 km.
3) Execute an appropriate SPARQL query. In case of the duty trip based advisory [5] this query should limit sought objects only to *ContactPerson* class instances (obviously, in the general case, such a query could seek other entities, e.g. golf courses, or historic castles).

   An additional SPARQL filter is applied according to the respective *Matching Criteria* part: *onto:locatedAt temp:gisResults-multi*. The matching request processing engine transforms the GIS sub-query results to a valid SPARQL filter and executes the query. In our example the final SPARQL query has the following form:

```
PREFIX onto:
<http://rossini.ibspan.waw.pl/
            Ontologies/KIST/KISTVO>
SELECT ?person
WHERE {?person isa onto:ContactPerson.}
FILTER (onto:locatedAt :RovaniemiCity).
```

   In our example, results of this query are *ContactPerson#1* and *ContactPerson#2* object references. Note that the proposed order of the GIS and the SPARQL query execution may change in the final version of our system, as it largely depends on results of our experiments with designing optimal SPARQL support for the, described above, GIS calculations.
4) Having merged results returned by the GIS component and the SPARQL engine, the relevance can be calculated. Note that the above proposed threshold value $R = \frac{1}{40}$ is a sample value only and is used to illustrate the process; an actual value will be a result of experimental calibration of the system. Specifically, to be able to actually establish a reasonable threshold value, a number of experiments have to be performed. Such experiments require a complete implemented system running and providing explicit and/or implicit user feedback. However, the question of tuning the performance of the proposed approach to a given institution is out of scope of this contribution. Thus the matching process involves:
   a) source instance $URI = DTR\#1$
   b) target objects $URI's = [ContactPerson\#1, ContactPerson\#2]$
   c) relevance threshold: $R = \frac{1}{40}$.

   If the relevance value for any object is above the relevance threshold, such object(s) will be suggested.

### A. Calculating relevance

Let us start from considering Figure 2 which presents the overview of relations between *Employee1* and two contact persons *ContactPerson#1* and *ContactPerson#2* via three research fields: *Volcanology*, *Paleontology*, *Geochronology*. These relations are represented in our ontology, as shown in Figure 3 (note that a figure which would include all relations would be too complex to be explanatory).

Figure 2 includes scaling factors (related to professional interests of the *Employee#1*), represented as $W1$, $W2$, and $W3$, which in the proposed algorithm, are used in order to calculate relevance between objects. The values of $D$ represent the ontology property weights for each relation, defined by an annotation property `voPropertyWeight`. On the other hand, Figure 3) presents in some detail links between the *Employee#1* and *ContactPerson#1* resources depicted from the perspective of ontology concepts.

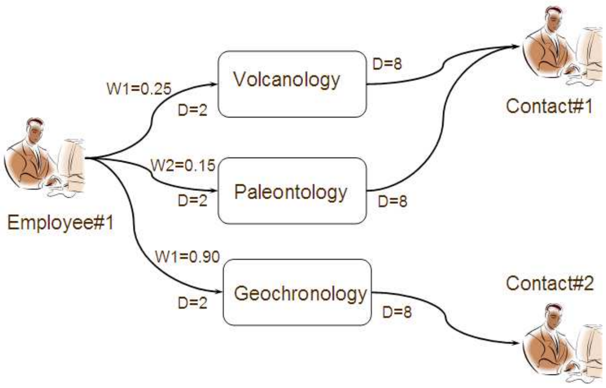


Fig. 2. Employee and foreign contacts

Therefore, based on Figure 2 and the above discussion, we can define three paths from *Employee#1* to the selected contact persons:

*Path* 1: *Employee#1* → *Employee#1Profile* → *Volcanology* → *ContactPerson#1Profile* → *ContactPerson#1*
*Path* 2: *Employee#1* → *Employee#1Profile* → *Paleontology* → *ContactPerson#1Profile* → *ContactPerson#1*
*Path* 3: *Employee#1* → *Employee#1Profile* → *Geochronology* → *ContactPerson#2Profile* → *ContactPerson#2*

Let us assume (see above) that ontology property weights are defined as follows:

$$\begin{aligned} voPropertyWeight(doesResearchInFields) &= 2\\ voPropertyWeight(isResearchedBy) &= 8\\ voPropertyWeight(hasProfile) &= 1\\ voPropertyWeight(belongsToResource) &= 1 \end{aligned}$$

Now, let us recall the fact that the *Employee#1* has different level of knowledge of these research fields, and the knowledge level (i.e. *weight*) can be applied to the `voPropertyWeight(doesResearchInFields)` value (i.e. *distance*) to obtain a scaled distance value for each individual. Precisely, the scaled relevance value is obtained by multiplying the individual weight value by the inverse of the distance value (i.e. the relevance value):

$$newRelevance = relevance \times weight$$

Since the *relevance value* between two nodes is the inverse of the *distance value*, the new distance is as follows:

$$newDistance = (\frac{1}{distance} \times \texttt{weight})^{-1}$$

This scaling provides personalized relevance results for the *Employee#1*. Here, scaled distance values from *Employee#1* to the three research fields (*Volcanology*, *Paleontology*, *Geochronology*) are 8, $\frac{40}{3}$, $\frac{20}{9}$, respectively. Now, the relevance value for each path is calculated as follows:

$$Rel_{path} = (\sum_{k=1}^{n}(k \times D_k))^{-1},$$

where $n$ is the number of edges in the path and $D_k$—distance of $k$-th edge. Thus the relevance result for each path is:

$$\begin{aligned} Rel_{path1} &= \frac{1}{45} = 0.022\\ Rel_{path2} &= \frac{3}{167} = 0.018\\ Rel_{path3} &= \frac{9}{301} = 0.030 \end{aligned}$$

Both $Rel_{path1}$ and $Rel_{path2}$ represent the relevance between *Employee#1* and *ContactPerson#1*, hence we can establish the final relevance value between the two objects by adding relevances of each path. Therefore, the final value of relevance between *Employee#1* and the two contact persons is:

$$\begin{aligned} &Rel_{ContactPerson\#1} = Rel_{path1} + Rel_{path2} = 0.040,\\ &Rel_{ContactPerson\#2} = 0.030. \end{aligned}$$

Note that the calculation process is presented in a simplified way in this paper, however, the detailed algorithm can be found in [16], [17], [20]. Based on the result and the proposed *Criteria*=$\{R \geq \frac{1}{40}$, where $R$ is the relevance threshold$\}$, both *ContactPerson#1* and *ContactPerson#2* will be recommended for *Employee#1*. Let us stress that *ContactPerson#2* is recommended via a single research field (*Geochronology*) whereas *ContactPerson#1* is recommended via combined strength of two research fields, even though the relevance value of each single path is below the threshold. Thus, the relevance measure considers not only a single research field match but also the "multidisciplinary" case. In the future we may consider making the threshold for the multi-pathway relevance to be different than the single-pathway one; as a single strong link is more important than a number of weaker links, but this will be done as a part of system calibration. Overall, as a result of the matching, the following additional duty trip activity is to be suggested:

```
:AdditionalDuty\#1 a onto:ISTDuty;
        onto:destination geo:RovaniemiCity;
        onto:madeContact :ContactPerson\#1.
:AdditionalDuty\#2 a onto:ISTDuty;
        onto:destination geo:RovaniemiCity;
        onto:madeContact :ContactPerson\#2.
:DTRProfile\#1 onto:duty :AdditionalDuty\#1.
:DTRProfile\#1 onto:duty :AdditionalDuty\#2.
```

The *:AdditionalDuty#1* and *:AdditionalDuty#2* objects define activities which are suggested to Mr. Chan who is planning his duty trip represented in the system as the *DTR#1*. Figures 3 and 4 present relations between objects included in the example in this section. Note that these figures omit *ContactPerson#2* and *AdditionalDuty#2* due to the fact that relations of these resources with *Employee#1* are analogical to the relation between *Employee#1* and *ContactPerson#1* and *AdditionalDuty#1*. In Figure 3 we present a path between the *Employee#1* and the *ContactPerson#1* and including them would only make both figures less legible. Finally, in Figure 4 we depict the final relations between the *Employee#1*, the *ContactPerson#1* and the *DutyTrip#1*, after the suggested *AdditionalDuty* is accepted.

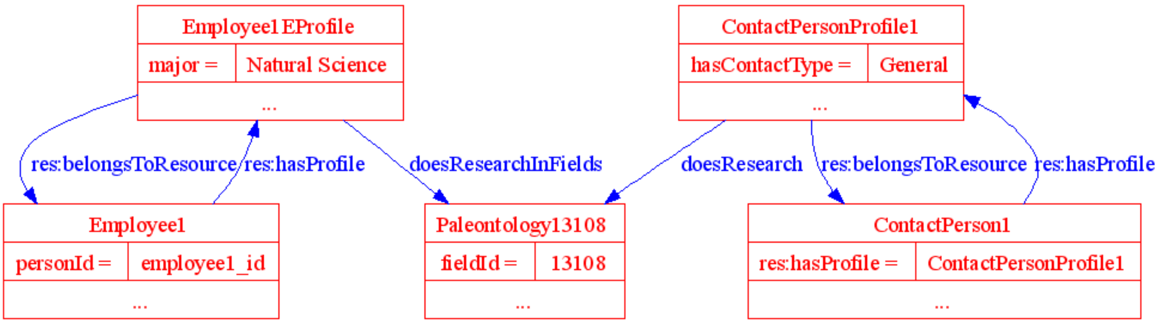


Fig. 3. Employee and foreign contact

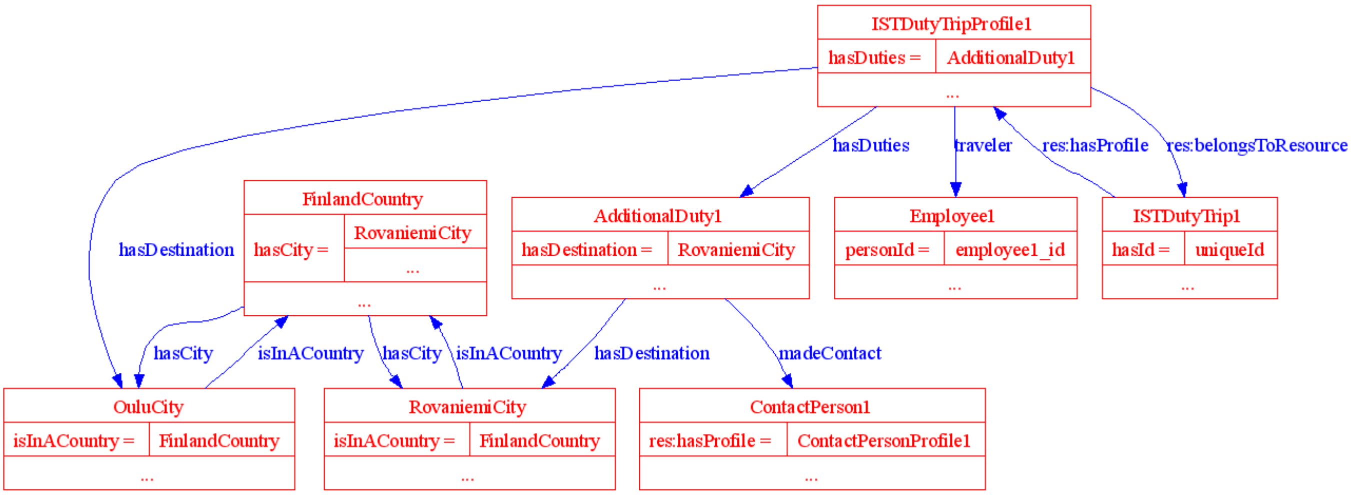


Fig. 4. Duty Trip related objects

## V. System building blocks

In [7] we have described in some details two main building blocks of the system: the *Relevance Calculation Engine* and the *Relevance Calculation Interface*. Therefore, here, we discuss only the *GIS subsystem*.

### A. The GIS Subsystem

In [6], [5], [7] we have outlined utilization of the GIS module—it is queried in case objects which have geospatial location properties that are involved in the matching operation. The state of the art research has shown that we can provide a reliable geospatial backend for our system by using the following components: (1) the GeoMaker [21] for collecting geographic coordinates of cities in the world, (2) the PostgreSQL database [22] for storing that information and for caching the result, and (3) Java *GIS—coordinates and distance cache* for calculating distance between cities, populating distance calculation results cached in the PostgreSQL database and interfacing the GIS module with the rest of the system. In Figure 5, which represents the GIS subsystem, these elements are placed on its bottom.

In the system that is supposed to communicate with the *GIS component*, the *GIS data consumer* is an interface that is responsible for requesting new data from the *GIS component*. On the system side, data is going to be stored in the semantic data storage that is based on the *JENA Model* [23]. We assume that the semantic data storage and *GIS component* share city instance identifiers in order to communicate in an optimal way in terms of performance.

For the purpose of calculating distance between two cities we utilize an implementation of the *Great Circle Distance Formula* [24]. This formula uses spherical trigonometry functions. Although relatively high precision of this method is not required in the system for the purpose of calculating distance between cities we apply it because the distances are calculated only once for each pair of cities.

$$\begin{aligned} result = 69.1 * \frac{180}{\pi} * \arccos\Big(\sin LAT1 * \sin LAT2 + \\ + \cos LAT1 * \cos LAT2 * \cos(LONG2 - LONG1)\Big) \end{aligned}$$

As it was pointed in previous sections, the *GIS component* is still under development and changes in its design (performance-related changes in particular) may be introduced as a result of further experiments.

## VI. Concluding remarks

In this paper we have presented a novel ontological matching algorithm in which we have combined matching based on

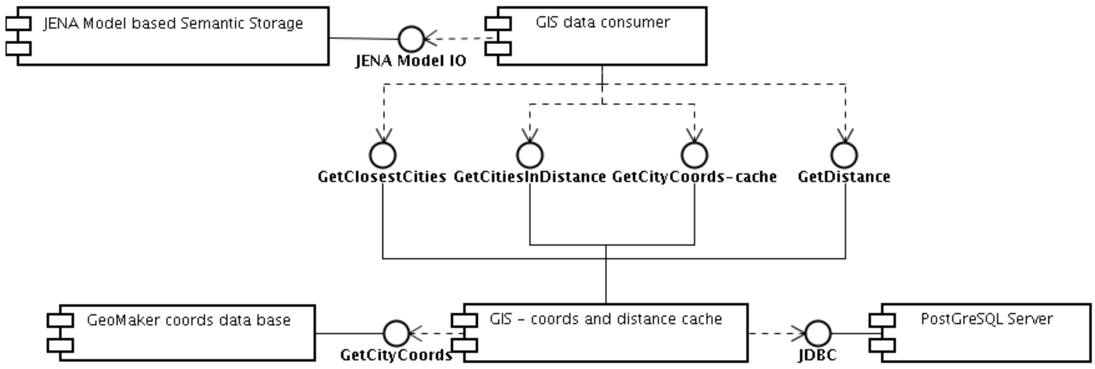


Fig. 5. The GIS subsystem; UML component diagram

ontological distance with filtering based on individual profiles. The proposed algorithm was illustrated in the context of a *Duty Trip Support* application. Across the paper we have identified a number of research questions, especially those related to an efficient implementation of geospatial data processing. We are currently implementing the proposed algorithm and the GIS subsystem as a part of the *DTS* application. Completing the initial implementation will allow us to start experimentally investigating all efficiency related questions. We will report on our progress in subsequent publications.

## Acknowledgment

This work is partially sponsored by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant.

## References

[1] http://en.wikipedia.org/wiki/Virtual_enterprise, 2008.

[2] http://www.businessdictionary.com/definition/virtual-organization.html, 2008.

[3] M. Ganzha, M. Gawinecki, M. Szymczak, G. Frackowiak, M. Paprzycki, M.-W. Park, Y.-S. Han, and Y. Sohn, "Generic framework for agent adaptability and utilization in a virtual organization—preliminary considerations," in *Proceedings of the 2008 WEBIST conference*, J. Cordeiro *et al.*, Eds. INSTICC Press, 2008, pp. IS–17–IS–25.

[4] M. Szymczak, G. Frackowiak, M. Gawinecki, M. Ganzha, M. Paprzycki, M.-W. Park, Y.-S. Han, and Y. Sohn, "Adaptive information provisioning in an agent-based virtual organization—ontologies in the system," in *Proceedings of the AMSTA-KES Conference*, ser. LNAI, N. Nguyen, Ed., vol. 4953. Heidelberg, Germany: Springer, 2008, pp. 271–280.

[5] G. Frackowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, M. Szymczak, M.-W. Park, and Y.-S. Han, *Considering Resource Management in Agent-Based Virtual Organization*, ser. Studies in Computational Intelligence. Heidelberg, Germany: Springer, 2008, in press.

[6] ——, "On resource profiling and matching in an agent-based virtual organization," in *Proceedings of the ICAISC'2008 conference*, ser. LNCS. Springer, 2008.

[7] M. Szymczak, G. Frackowiak, M. Ganzha, M. Paprzycki, M.-W. Park, Y.-S. Han, Y. T. Sohn, J. Lee, and J. K. Kim, "Infrastructure for ontological resource matching in a virtual organization," in *Proceedings of the IDC Conference*, ser. Studies in Computational Intelligence, N. Nguyen and R. Katarzyniak, Eds., vol. 134. Heidelberg, Germany: Springer, 2008, pp. 111–120.

[8] M. Ganzha, M. Paprzycki, M. Gawinecki, M. Szymczak, G. Frackowiak, C. Badica, E. Popescu, and M.-W. Park, "Adaptive information provisioning in an agent-based virtual organization—preliminary considerations," in *Proceedings of the SYNASC Conference*, ser. LNAI, N. Nguyen, Ed., vol. 4953. Los Alamitos, CA: IEEE Press, 2007, pp. 235–241.

[9] M. Szymczak, G. Frąckowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, and M.-W. Park, "Resource management in an agent-based virtual organization—introducing a task into the system," in *Proceedings of the MaSeB Workshop*. Los Alamitos, CA: IEEE CS Press, 2007, pp. 458–462.

[10] "Korea science and engineering foundation," http://www.kosef.re.kr/english_new/index.html.

[11] C. Badica, E. Popescu, G. Frackowiak, M. Ganzha, M. Paprzycki, M. Szymczak, and M.-W. Park, "On human resource adaptability in an agent-based virtual organization," in *New Challenges in Applied Intelligence Technologies*, ser. Studies in Computational Intelligence, R. K. N.T. Nguyen, Ed., vol. 134. Heidelberg, Germany: Springer, 2008, pp. 111–120.

[12] H. S. Pinto and J. P. Martins, "Ontology integration: How to perform the process," in *Proceedings do Workshop Ontologies and Information Sharing, realizado durante a conferência International Joint Conference in Artificial Intelligence, IJCAI2001*, Seattle, Washington, USA, 2001, pp. 71–80.

[13] A. Gangemi, D. M. Pisanelli, and G. Steve, "Ontology integration: Experiences with medical terminologies," in *Proceedings of Formal Ontology in Information Systems, FOIS'98*, N. Guarino, Ed. IOS Press, 1998, pp. 163–178.

[14] J. Euzenat and P. Shvaiko, *Ontology Matching*, ser. Studies in Computational Intelligence. Heidelberg, Germany: Springer, 2007.

[15] "Sparql query language for rdf," http://www.w3.org/TR/rdf-sparql-query.

[16] S. Rhee, J. Lee, and M.-W. Park, "Ontology-based semantic relevance measure," *CEUR-WS*, vol. 294, no. 1613–0073, 2007.

[17] ——, "Riki: A wiki-based knowledge sharing system for collaborative research projects," in *Proceedings of the APCHI 2008 Conference*, ser. LNCS. Springer, 2008.

[18] "Extensible sparql functions with embedded javascript," http://online-journals.org/proceedings/article/view/232/164.

[19] G. Frąckowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, M. Szymczak, C. Bădică, Y.-S. Han, and M.-W. Park, "Adaptability in an agent-based virtual organization," *Internetional Journal Accounting, Auditing and Performance Evaluation*, 2008, in press.

[20] S. Rhee, J. Lee, and M.-W. Park, "Semantic relevance measure between resources based on a graph structure," in *Proceedings of the IMCSIT'08 Conference*, 2008, in press.

[21] "Geomaker," http://pcwin.com/Software_Development/GeoMaker/index.htm.

[22] "Postgis:home," http://postgis.refractions.net/.

[23] "Jena—a semantic framework for java," http://jena.sourceforge.net, 2008.

[24] http://www.meridianworlddata.com/Distance-calculation.asp, 2008.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Computer-aided Detecting of Early Strokes and its Evaluation on the Base of CT Images

Elżbieta Hudyma
Institute of Applied Informatics Wroclaw University
of Technology Wybrzeze Wyspianskiego 27,
Wroclaw, Poland
Email: elzbieta.hudyma@pwr.wroc.pl

Grzegorz Terlikowski
Institute of Applied Informatics Wroclaw University
of Technology Wybrzeze Wyspianskiego 27,
Wroclaw, Poland
Email: terlikowski.grzegorz@gmail.pl

***Abstract*—This paper presents an easy way of finding strokes on computer tomography images. By calculating a cohesive rate (CR) of suspicious pixels on a series of CT images there is a possibility of calculating a general probability of a stroke. In a difficult case there is always an opportunity to generate a graph of all stroke probabilities interposed on the original image. It is a very helpful tool for specialists and neurologists working in emergency situations . Supported by grant N518 022 31/1338 (Ministry of Science).**

## I. Introduction

A STROKE is a rapidly developing loss of brain functions due to a disturbance in the blood vessels supplying the brain. This can be due to ischemia (lack of blood supply), or due to a hemorrhage which is caused by a blood vessel that breaks and bleeds into the brain. The most common kind an is ischemic stroke. It is /found in about 85% of all the patients with strokes. This kind of disease is very serious and without suitable treatment it leads to death or long term disability.

The availability of treatments, when given at an early stage, can reduce stroke severity. Early diagnosis of acute cerebral infraction is critical due to the timing of thrombolytic treatment.

The clinical diagnosis of an ischemia stroke is difficult and it has to be supported by brain imaging. There are new effective methods for detecting strokes based on the images, but in most cases, a CT remains the most important and the most popular brain imaging tool. It is vital that no longer than 3 hours elapse between diagnosis and action planning that the appropriate treatment is given. During the initial 3 hours the area's ischemia CT attenuation decreases by 2-3 HU (Hunsfield Unit)[1]. The distinction of the colors on the CT images is so small that even a neurologist with great experience cannot see it.

After a few days damaged tissue is very noticeable, but it is too late for effective treatment. Because of the great importance of an early detection of the stroke, many authors presented various approaches to that problem [2-4].

The aim of this paper is to present a self-constructed, effective algorithm to support ischemic stroke detection based on CT images.

## II. Algorithm

The proposed heuristic algorithm analyzes a series of CT images of a brain. As a result algorithm generates graphs of the suspicious areas of the brain. Parts of the brain tissue which have the biggest probability of a stroke are highlighted. There is also a possibility to generate a full graph which contains the probabilities of a stroke in a colorful version. The algorithm also calculates a general probability of a stroke. Calculation are based on the patient's CT images. The method is based on a few important statements.

The first of them is about colors which represent the area of the probable stroke's location. According to [2] the area of ischemia CT attenuation decreases by 2-3 HU which means that the suspicious part in the CT image are a little darker than the common tissue of the brain.

The second statement is that strokes have a volume. It gives a possibility of seeing a stroke on more than one image.

The next statement is that strokes are solid structures. The lack of blood supply on the CT image looks like a large, solid and rounded figure.

The probability of a stroke happening on both sides of the brain is almost zero. The algorithm uses this fact and assumes that a stroke can be found only on one side of the brain[1].

The stroke detection contains four main stages.

### A. Preprocessing

An image has to be prepared by separating the brain tissue from the scan, finding a symmetry of the brain and selecting suspicious points of the CT image where a stroke can be found. . Preprocessing is extremely important because it shows significant features of the analyzed case.

Firstly, non-brain tissue must be removed from a CT image. It may be done through region growing. The skull surrounding the brain has an uninterrupted, rounded shape and on image it has a white color. That is why extraction of the brain tissue can be done by using simple figure filling algorithm (e.g., Smith's algorithm[5]).The starting point for extraction algorithm is the center of image and the filled colors have to be other than white (the skull is represented by a white color).

978-83-60810-14-9/08/$25.00 © 2008 IEEE

The second preparation step is to find a color which is the most common in the image. Using this information the suspicious point of the image can be found. Creating a histogram H where $i \in \langle 0;255 \rangle \wedge i \in N$ and $H_i$ is a number of pixels where color is *i* makes it easy to find and set $C_{max}$ as the most common color [6].

With this information the suspicious pixels are easily found. According to statements and experimental study colors which may represent a stroke are in range $K \in \langle C_{\max} - g ; C_{\max} \rangle$ where g is a precision property.

Searching the symmetry line of the brain is a really important step. It can be found by rotating the image and looking for two parallel straight lines with the smallest distance between them. The $J_R$ and $J_L$ contain pixels which are on the right and left side of the brain.

Using the previous calculation suspicious areas of pixels can be found. Taking into consideration the symmetry line and points which have colors from the range K, two subsets $T_R \subset J_R$ and $T_L \subset J_L$ are created.

### B. Cohesive rate(CR)

The main part of the algorithm is to calculate the cohesive rate of suspicious pixels. CRs value represents the summary relative locations of one suspicious pixel to the rest of them. It is the key to this algorithm. CR is defined by a formula:

$$\forall\, p \in T_V \left( cohesive_{rate}(p) = \sum_{i=1}^{T_V} \left| 1/distance(p, p_i) \right| \right)$$ where

V={R,L}. The distance means distance in Manhattan's metric. Value $P_{max}$ is set as maximum cohesive rate from both $T_V$ subsets. According to statements the stroke is a solid structure which means that CR gets high values for stroke pixels.

### C.Probability of a stroke

The algorithm calculates a general probability of a stroke for a series of CT images by taking under consideration the single scans stroke risk.

For selected $k \in (0;1)$ calculate number of pixels $U_V$ which cohesive rate is from range (k $P_{max}$; $P_{max}$) and are in set $T_V$, where V={R,L}. *k* is a kind of a sensitivity property. A bigger k causes a bigger probability of a stroke. On the other hand a minor k causes a lesser probability of a stroke

According to statements and taking into consideration that stroke is only on one side of the brain the probability of a stroke for left and right side can be calculated by a formula: $P_V=U_V/(U_R+U_L)$, where V={R,L}. The formula which describes calculations uses the information about number of selected pixels generated is previous step. Using this calculations it is easy way to estimate general stroke risk.

Simply taking all $P_R$ and $P_L$ for a series of CT images and calculating average values of them gives the probability of a stroke for left $P_{AVGL}$ and right $P_{AVGR}$ side of the brain.

General probability of a stroke for a series of CT images is defined by a formula: $P = \dfrac{|P_{AVGR} - P_{AVGL}|}{P_{AVGR} + P_{AVGL}}$ .

### D. Visualization

Visualization [6] is the final step which is not necessary but extremely important in difficult cases. There are two types of visualization. The first is to generate an original image with selected pixels by sensitivity factor. The second type shows all suspicious pixels but the color of the pixels depends on the CR for that point. For this one there is no need to specify sensitivity property.

## III. Experimental Study

Experimental tests were performed to verify the efficiency of the algorithm for the various values of parameters g and k. Tests were carried out on 23 CT images specially selected to represent different positions of the stroke in the brain as well as images of a healthy brain. Among the images chosen there are 3 series of scans and the rest are single representations. Results were very optimistic. First tests revealed the potential of the algorithm, the marking was clear and accurate. Sometimes it seemed to show details that were not noticeable in the original image. After comparison of processed images of a healthy brain and one with a stroke the difference was obvious even for someone with no experience in this kind of analysis.

All of the strokes were pointed correctly with their placement and size. As an example 3 CT images were chosen (Fig. 1) and tested with different parameters. The first pair presents a healthy brain. To show how it works in different cases other two pairs of the images show a brain attacked by the disease.

Every presented case contains a table with algorithm results for the various sensitivity and precision Parameters. There is also a figure of visualizations for each factor combination. The shortcuts used in the tables mean:

*I. g – precision factor,*
*II.k – sensitivity property,*
*III.* $U_L$ *– number of selected pixels on the left side*
*IV.* $U_P$ *– number of selected pixels on the right side*
*V.* $CR_L$ *– maximum cohesive rate for the left side*
*VI.* $CR_R$ *– maximum cohesive rate for the right side*

Fig 2. shows screenshots of a healthy brain with different parameters values set. For different factors the algorithm calculates the probability of a stroke. All of the calculated values are combined in Table I and for all of the different values set the probability is very low. It is because cohesive rates for left and right side are almost equal which leads to a conclusion that there are no suspicious areas.

Following images in Fig. 3 and Fig. 4 are very difficult for neurologists to analyze. Strokes are hard to notice because the difference between pixels representing the stroke and normal tissue is very small. This algorithm helps to separate the stroke by finding even a small change between pixels and calculating cohesive rate. Table II and III show calculated values due to differently set parameters. Unlike previous analysis these two cases show high probability of a stroke in every combination of the variables. There are certain adjustments that make the probability go high, even up to 1,00. In both cases the algorithm has highlighted the suspicious areas and clearly announced that a stroke has been found in the image.

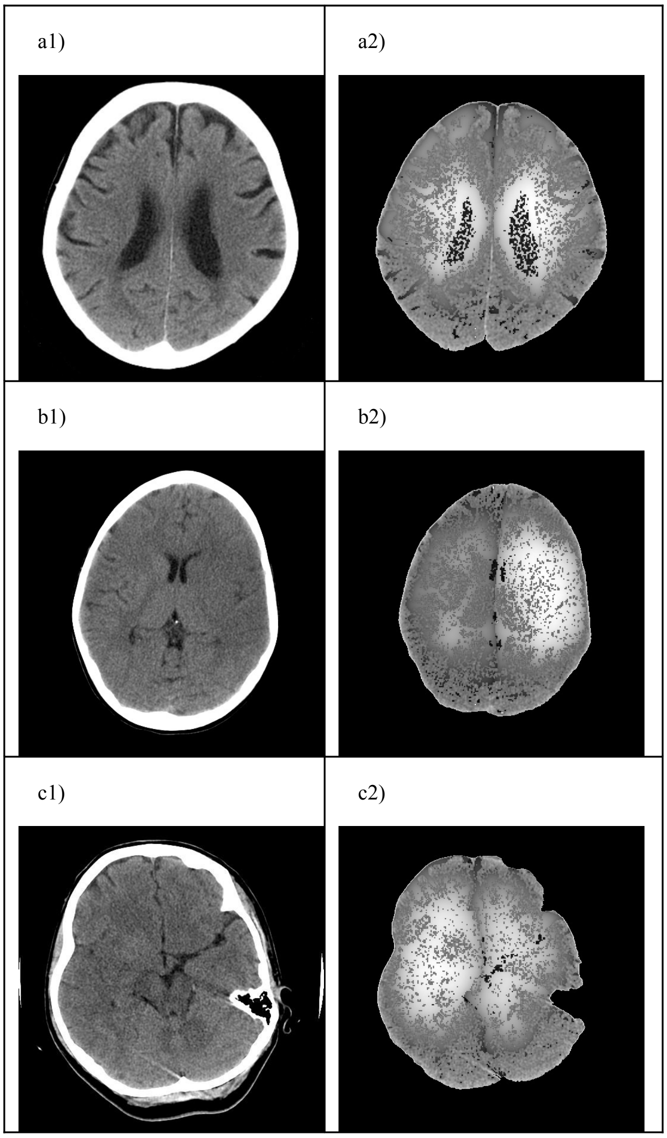


Fig. 1 Pairs of original (left) and fully visualized (right) CT images.

Table I.
Algorithm results for the various sensitivity and precision factor for healthy brain

| g | k | $U_L$ | $U_R$ | $CR_L$ | $CR_R$ | Fig | Probability |
|---|---|---|---|---|---|---|---|
| 20 | 0,95 | 178 | 212 | 55 | 54 | 2.a) | 0,09 |
| 20 | 0,90 | 433 | 527 | 55 | 54 | 2.b) | 0,10 |
| 20 | 0,80 | 1194 | 1238 | 55 | 54 | 2.c) | 0,02 |
| 20 | 0,70 | 1918 | 1892 | 55 | 54 | 2.d) | 0,01 |
| 40 | 0,90 | 744 | 893 | 70 | 69 | 2.e) | 0,09 |
| 60 | 0,90 | 1058 | 1410 | 79 | 79 | 2.f) | 0,14 |

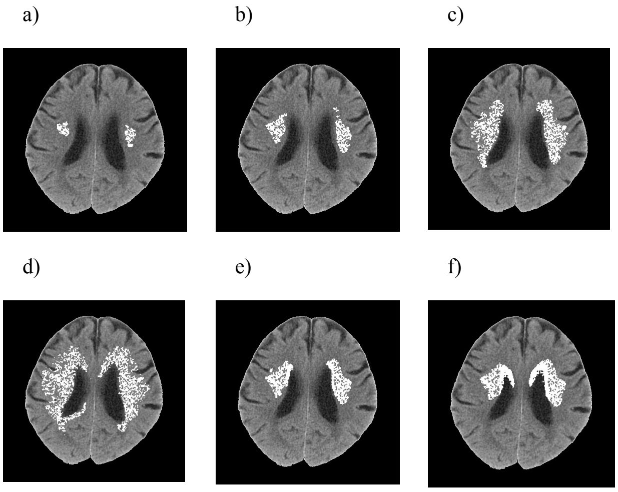


Fig. 2 Visualization for Table I – healthy brain.

Table II.
Algorithm results for the various sensitivity and precision factor for first difficult case

| g | k | $U_L$ | $U_R$ | $CR_L$ | $CR_R$ | Fig | Probability |
|---|---|---|---|---|---|---|---|
| 20 | 0,95 | 0 | 778 | 58 | 76 | 3.a) | 1,00 |
| 20 | 0,90 | 0 | 1637 | 58 | 76 | 3.b) | 1,00 |
| 20 | 0,80 | 0 | 2952 | 58 | 76 | 3.c) | 1,00 |
| 20 | 0,70 | 687 | 3877 | 58 | 76 | 3.d) | 0,70 |
| 40 | 0,90 | 0 | 2492 | 65 | 90 | 3.e) | 1,00 |
| 60 | 0,90 | 0 | 2622 | 69 | 92 | 3.f) | 1,00 |

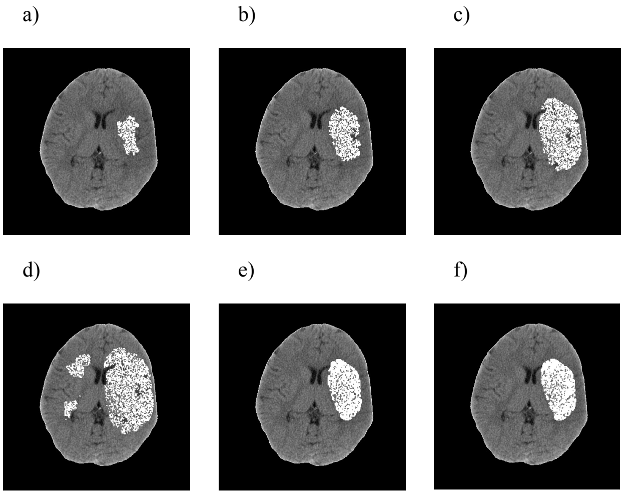


Fig. 3 Visualization for Table II – stroke on the right side.

TABLE III.
ALGORITHM RESULTS FOR THE VARIOUS SENSITIVITY AND PRECISION FACTOR FOR SECOND DIFFICULT CASE

| g | k | $U_L$ | $U_R$ | $CR_L$ | $CR_R$ | Fig | Probability |
|---|---|---|---|---|---|---|---|
| 20 | 0,95 | 788 | 0 | 60 | 50 | 4.a) | 1,00 |
| 20 | 0,90 | 1618 | 0 | 60 | 50 | 4.b) | 1,00 |
| 20 | 0,80 | 2809 | 234 | 60 | 50 | 4.c) | 0,85 |
| 20 | 0,70 | 3596 | 1367 | 60 | 50 | 4.d) | 0,45 |
| 40 | 0,90 | 2084 | 0 | 94 | 80 | 4.e) | 1,00 |
| 60 | 0,90 | 2187 | 0 | 97 | 86 | 4.f) | 1,00 |

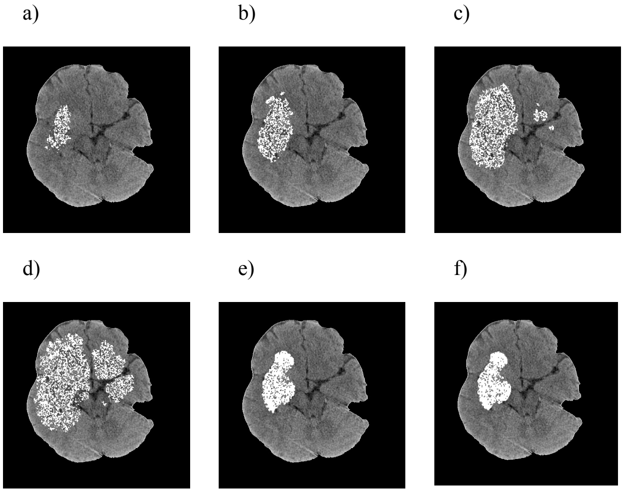


Fig. 4 Visualization for Table III – stroke on the left side.

For the sensitivity factor set to 0.9 and precision property to 20, average probability of a stroke in 8 series has been calculated and it is 0,70%. Standard deviation for these cases is 0,20. It is because some of the CT images contains very dark strokes and for precision property set to 20 it is hard to find it. Changing it to 60 or higher gives better results.

## IV. CONCLUSIONS AND DISCUSSION

The presented method gives satisfying results. The quality of results given by the algorithm and its performance can be set by the sensitivity factor and the precision parameter. The optimal values were found experimentally by analyzing series of CT images. The best settings are 0.9 for the sensitivity factor and 20 for the precision parameter. The precision parameter is most important for the performance of the method and has a minor influence on quality which can be changed by adjusting the sensitivity factor. Although both parameters can be set manually, it is better to use their optimal settings or to change them only in difficult cases to estimate the volume of a stroke

The presented test sets do not include series. It is because it will take up much space. The series were tested on 11 people with strokes and on 15 healthy people and results were very optimistic. The probability for different cases were from 0.6 to 0.95 for CT images with a stroke and 0.0 to 0.5 without a , stroke, the sensitivity factor having been set at 0.9 and precision property at 20.

Another interesting option is that a cohesive rate graph can be displayed and it can be useful for additional analysis of the brain. There is an example of the graph in this paper shown in grayscale instead of a colored one. In the colored graph the highlighted areas have different colors and intensity. Colors are specially chosen to attract human eyes. Red represents the areas which have the biggest probability of being a stroke, green marks the areas with medium danger and blue is reserved for the less suspected parts of the brain. Thanks to this color palette the graph shows a general view of the brain tissue separated from any distractions from the original image.

There are possibilities to improve this algorithm. Better optimalization can be done by rewriting it in Assembler. There is also an idea to use folds analysis by using the fact that in the area where the stroke is the folds are smaller or not noticeable. The changes proposed here can be very useful to improve the algorithm effectivness and performance.

New discoveries can bring new facts to light about the anatomy and characteristic of a stroke. They can also improve the method by taking them into consideration if possible. There is no possibility for a computer program to replace a good specialist but it can be useful for faster diagnosis because it shows by means of a number the probability of a stroke.

## V. REFERENCES

[1] http://www.strokecenter.org/prof/
[2] A. Przelaskowski, K. Sklinda, G. Ostrek,E: Stroke Display Extensions: Three Forms of Visualization, in "Information Tech. In Biomedicine", 2008 Springer-Verlag Berlin Heidelberg, pp. 139-148, 2008
[3] J. L. Starck, F. Murtagh, E. J. Candes, D. L. Donoho: Grey and Color Image kontrast enhancement by the curvelet transform. IEEE Trans. Image Proc. 12(6), 706-717 (2003)
[4] N. Bonnier, E. P. Simoncelly: Locally adaptivemulticale kontrast optimization. Proc. IEEE ICIP 2 1001-1004 (2005).
[5] http://grafika.aei.polsl.pl/doc/01-RASb.pdf
[6] Ch. D. Watkins, A. Saudun, S. Marenka: Nowoczesne metody przetwarzania obrazu, WNT, 1995

# Computer Aspects of Numerical Algorithms

NUMERICAL algorithms are widely used by scientists engaged in various areas. There is a special need of highly efficient and easy-to-use scalable tools for solving large scale problems. The workshop is devoted to numerical algorithms with the particular attention to the latest scientific trends in this area and to problems related to implementation of libraries of efficient numerical algorithms. The goal of the workshop is meeting of researchers from various institutes and exchanging of their experience, and integrations of scientific centers.

## Topics

- Parallel, distributed and grid numerical algorithms
- Data structures for numerical algorithms
- Libraries for numerical computations
- Numerical algorithms testing and benchmarking
- Analysis of rounding errors of numerical algorithms
- Optimization of numerical algorithms
- Languages, tools and environments for programming numerical algorithms
- Paradigms of programming numerical algorithms
- Contemporary computer architectures
- Applications of numerical algorithms in science and technology

## International Programme Committee

**Krzysztof Banaś,** AGH University of Science and Technology, Poland

**Vasile Berinde,** North University of Baia Mare, Romania

**Luigi Brugnano,** Universita degli Studi, Firenze, Italy

**Stefka Dimova,** Sofia University St. Kliment Ohridski, Bulgaria

**Domingo Giménez Cánovas,** University of Murcia, Spain

**George A. Gravvanis,** Democritus University of Thrace, Greece

**Tadeusz Czachórski,** The Institute of Theoretical and Applied Informatics, PAS

**Tim Davis,** University of Florida, USA

**Graeme Fairweather,** Colorado School of Mines, USA

**Mauro Francaviglia,** Dipartimento di Matematica, Torino, Italy

**Salvatore Filippone,** Universita' di Roma "Tor Vergata", Italy

**Ian Gladwell,** Southern Methodist University, USA

**Grigoris Kalogeropoulos,** University of Athens, Greece

**Jerzy Klamka,** The Institute of Theoretical and Applied Informatics, PAS

**Jerzy Kozicki,** Maria Curie-Sklodowska University, Poland

**Stanislaw Kozielski,** Silesian University of Technology, Poland

**Anna Kucaba-Pietal,** Rzeszow University of Technology, Poland

**Vyacheslav Maksimov,** IMM RAS and Ural State Technical University, Russia

**Dana Petcu,** Western University of Timisoara, Romania

**Fayaz R. Rofooe,** Sharif University of Technology, Tehran, Iran

**Bianca Satco,** Stefan cel Mare University, Suceava, Romania

**Vladimir Sergeichuk,** Institute of Mathematics, National Academy of Sciences, Ukraine

**T. E. Simos,** University of Peloponnese, Greece

**Dharmendra Sharma,** Faculty of Information Sciences and Engineering, University of Canberra, Australia

**Tanush Shaska,** Oakland University, Rochester, MI, USA

**Olga Shishkina,** Institute for Aerodynamics and Flow Technology, Germany

**Tomasz Szulc,** Adam Mickiewicz University, Poland

**Andrei I. Tolstykh,** Computing Center of Russian Academy of Scienses

**Marek Tudruj,** Institute of Computer Science Polish Academy of Sciences & Polish-Japanese Institute of Information Technology, Poland

**Vasyl Ustimenko,** Maria Curie-Sklodowska University, Poland

**Marian Vajtersic,** Salzburg University, Austria

**Krystyna Ziętak,** Wrocław University of Technology, Poland

## Organizing Committee

**Przemyslaw Stpiczynski, Beata Bylina and Jaroslaw Bylina,** Maria Curie-Sklodowska University, Poland

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 257 – 261

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Numerical Algorithm of Solving the Forced sine-Gordon Equation

Alexandre Bezen
School of Life and Physical Sciences, RMIT University, Melbourne, GPO Box 2476V, Melbourne VIC 3001,
Australia, Email: *abezen@unimelb.edu.au*, and Department of Mechanical and Manufacturing Engineering,
University of Melbourne

***Abstract*—The numerical method of solving the problem of small perturbations of a stationary traveling solution (soliton) of well-known in physics sin-Gordon equation is presented. The solution is reduced to solving a set of linear hyperbolic partial differential equations. The Riemann function method is used to find a solution of a linear PDE. The value of the Riemann function at any particular point is found as a solution of an ordinary differential equation. An algorithm of calculation of a double integral over a triangular integration area is given.**

## I. Introduction

THE RIEMANN method is an important technique for solving Cauchy problems for partial differential equations. However, it does not yield closed form solutions except in few cases of equations with constant coefficients [1]. A typical example of such an equation is the forced sine-Gordon equation

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} + \sin u = \varepsilon\Phi(t) \qquad (1)$$

where $\varepsilon = 1$ is a small parameter, $t$ is a time variable, $x$ is a space variable, and $\Phi(t)$ is a deterministic or random external force with known statistics. Equation (1) with zero right-hand side possesses stationary traveling solutions depending on a variable $\chi = x - Vt$ $(-1 < V < 1)$. The particular solution that is of great physical interest is the kink or soliton [2] (Fig. 1)

$$u_0(\chi) = 4\operatorname{atan}\left[\exp\left(\chi/\sqrt{a}\right)\right] \qquad (2)$$

where $a = 1 - V^2$. The problem of deterministic or stochastic perturbations of the kink solution is important in physical applications.

The approximate solution of the equation (1) with non-zero function $\Phi(t)$ can be constructed by the asymptotic method using a smallness of the parameter $\varepsilon$:

$$u(\chi,t) = u_0(\chi) + \varepsilon u_1(\chi,t) + \varepsilon^2 u_2(\chi,t) + \mathrm{K} \qquad (3)$$

where $u_0(\chi)$ is the solution (2). The functions $u_1, u_2, \ldots$ are solutions of linear hyperbolic equations.

Apparently, Walsh's papers [3], [4] were the most remarkable first investigation of stochastic processes described by the second order partial differential equations. In particular, a weak solution of a stochastic partial differential equation was defined as a solution of an integral equation. Walsh in [2] applied the Green's method to simple linear hyperbolic and parabolic equations in case of the white noise, and Carmona and Nualart in [5] proved that the weak solution exists and it is unique.

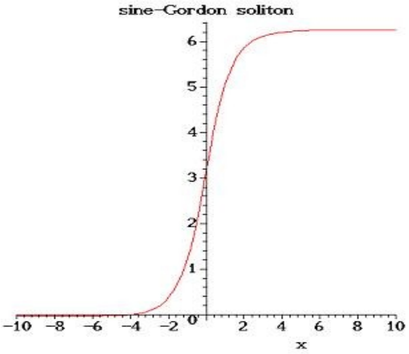


Fig. 1. The sin-Gordon soliton.

In this paper, the Riemann function method is used to find a solution of linear hyperbolic equations (4) and (5). The algorithm is numerical and can be divided into two parts. First, the Riemann function at each point of the integration area is found as a solution of an ordinary differential equation. Second, the triangular integration area is transformed into a rectangle. This allows simplifying and improving considerably calculation of a double integral. Initially the method was described in [6] and [7] and was applied in a case of stochastic perturbations. In the current paper, the proposed method has been improved. The physical part of the method and some numerical results were described in [8].

## II. The Solution Form of the Eequation (1)

In the new variables $(\chi, t)$ the equation (1) becomes

$$\frac{\partial^2 u}{\partial t^2} - 2V\frac{\partial^2 u}{\partial\chi\partial t} - a\frac{\partial^2 u}{\partial\chi^2} + \sin u = \varepsilon\Phi(t) \qquad (4)$$

Expanding $\sin u$ in power series in $\varepsilon$

$$\sin u(\chi,t)=\sin u_0(\chi)+\varepsilon u_1(\chi,t)\cos u_0(\chi)+$$
$$\varepsilon^2\left[u_2(\chi,t)\cos u_0(\chi)-\frac{u_1^2(\chi,t)}{2}\sin u_0(\chi)\right]+\ldots \qquad (5)$$

Substituting (3) and (5) into (4) in the first order on $\varepsilon$ we obtain

$$\frac{\partial^2 u_1}{\partial t^2}-2V\frac{\partial^2 u_1}{\partial\chi\partial t}-a\frac{\partial^2 u_1}{\partial\chi^2}+\cos u_0(\chi)u_1=\Phi(t) \qquad (6)$$

In the second order

$$\frac{\partial^2 u_2}{\partial t^2}-2V\frac{\partial^2 u_2}{\partial\chi\partial t}-\left(1-V^2\right)\frac{\partial^2 u_2}{\partial\chi^2}+\cos u_0(\chi)u_2=$$
$$\frac{1}{2}u_1^2\sin u_0(\chi) \qquad (7)$$

To apply the Riemann's method of solving (6) and (7) [1] we need to transform these equations to the standard form which does not contain the second mixed derivative. To get rid of the mixed derivatives let us make a transformation $\chi=\chi$ and $\tau=t-\dfrac{V\chi}{a}$. In the new variables equations (6) and (7) read

$$\frac{\partial^2 u_1}{\partial\tau^2}-a^2\frac{\partial^2 u_1}{\partial\chi^2}+a\cos u_0(\chi)u_1=a\sin\left(\tau+\frac{V\chi}{a}\right) \qquad (8)$$

and

$$\frac{\partial^2 u_2}{\partial\tau^2}-a^2\frac{\partial^2 u_2}{\partial\chi^2}+a\cos u_0(\chi)u_2=$$
$$\frac{1}{2}au_1^2\sin u_0(\chi) \qquad (9)$$

with trivial initial conditions over the straight line $C$

$$C:\tau=-\frac{V}{a}\xi \qquad (10)$$

## III. The Riemann Function

Equations (8) and (9) have the same left-hand side and can be presented as

$$\frac{\partial^2 u_1}{\partial\tau^2}-a^2\frac{\partial^2 u_1}{\partial\chi^2}+a\left[1-f(\chi)\right]u_1=a\Phi\left(\tau+\frac{V\chi}{a}\right) \qquad (11)$$

$$\frac{\partial^2 u_2}{\partial\tau^2}-a^2\frac{\partial^2 u_2}{\partial\chi^2}+a\left[1-f(\chi)\right]u_2=\frac{a}{2}u_1^2\sin u_0(\chi) \qquad (12)$$

where $f(\chi)=2/\cosh^2(\chi/\sqrt{a})$ since

$$\cos[4\arctan(e^{\frac{1}{\sqrt{a}}\chi})]=1-\frac{2}{\cosh^2(\frac{1}{\sqrt{a}}\chi)}$$

The graph of the function $f(\chi)$ for the value $V=0.5$ is shown in Fig. 2.

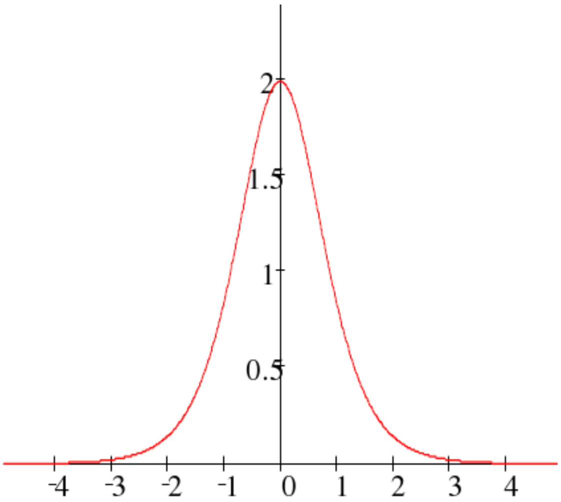


Fig .2. The function $f(\chi)$ , $a=1$ .

The corrections $u_i(\chi_0,\tau_0)$ ,$i=1,2$ to the kink solution $u_0(\chi_0)$ at a point $(\chi_0,\tau_0)$ can be presented through the Riemann function $\omega(\chi,\tau)$ [1]:

$$u_1(\chi_0,\tau_0)=a\iint_A\Phi\left(\tau+\frac{V\chi}{a}\right)\omega(\chi,\tau)d\chi d\tau \qquad (13)$$

$$u_2(\chi_0,\tau_0)=\frac{a}{2}\iint_A u_1^2(\chi,\tau)\sin u_0(\chi)\omega(\chi,\tau)d\chi d\tau$$

(14)

where $A$ is the characteristic triangle in the plane $(\chi,\tau)$, bounded by the straight line $C$ given in (10) and the characteristics WA$_{□}$ $:\tau\quad\frac{1}{a}(\chi-\chi_0)=\tau_0$ (Fig. 3).

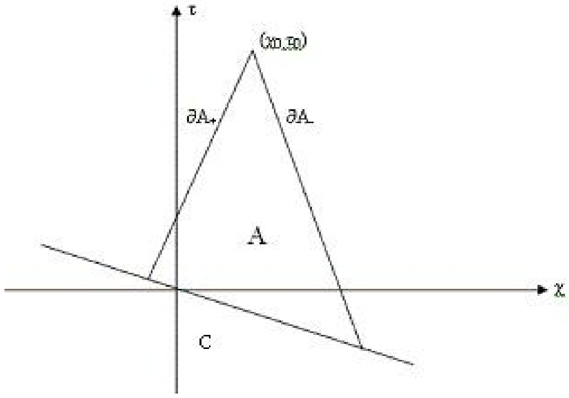


Fig. 3. The characteristic triangle $A$.

The Riemann function $\omega(\chi,\tau)$ satisfies the equation

$$\frac{\partial^2\omega}{\partial\tau^2}-a^2\frac{\partial^2\omega}{\partial\chi^2}+b^2(1-f(\chi))\omega=0 \qquad (15)$$

with $\omega=1/(2a)$ over the characteristics and $b=\sqrt{a}$ .

Since $f(\chi) \to 0$ outside of a finite interval $(-\chi_{\lim}, \chi_{\lim})$ then one can expect that the solution of (15) can be close to the solution of the telegrapher's equation [1]:

$$\frac{\partial^2 \omega}{\partial \tau^2} - a^2 \frac{\partial^2 \omega}{\partial \chi^2} + b^2 \omega = 0$$

which is $\omega(\chi, \tau) = \frac{1}{2a} J_0(b\gamma)$ and $\gamma = \sqrt{(\tau - \tau_0)^2 - \frac{1}{a^2}(\chi - \chi_0)^2}$ . The variable $\gamma$ is a hyperbolic distance in the characteristic triangle $A$ . The distance $\gamma$ is positive inside $A$, i.e. $(\tau - \tau_0)^2 > \frac{1}{a^2}(\chi - \chi_0)^2$ , and imaginary when $(\tau - \tau_0)^2 < \frac{1}{a^2}(\chi - \chi_0)^2$ . It vanishes when $(\tau - \tau_0)^2 = \frac{1}{a^2}(\chi - \chi_0)^2$ .

The equation (15) in the variables $(\chi, \gamma)$ has the form

$$\frac{\partial^2 \omega}{\partial \gamma^2} - a^2 \frac{\partial^2 \omega}{\partial \chi^2} + 2\frac{\chi - \chi_0}{\gamma} \frac{\partial^2 \omega}{\partial \chi \partial \gamma} + \frac{1}{\gamma}\frac{\partial \omega}{\partial \gamma} + \\ b^2(1 - f(\chi))\omega = 0 \qquad (16)$$

Let's look for the solution of (16) in the form

$$\omega(\chi, \gamma) = \frac{1}{2a} J_0(b\gamma) + \Phi(\chi)\sum_{n=1}^{\infty} A_n (b\gamma)^n \qquad (17)$$

where the first term of the sum is the solution of the telegrapher's equation and the second term must be small for a finite value of $\chi_{\lim}$ .

Since

$$J_0''(b\gamma) + \frac{1}{b\gamma} J_0(b\gamma) + J_0(b\gamma) = 0$$

then substitution of (17) into (16) gives

$$\Phi(\chi)\sum_{n=1}^{\infty} n(n-1)A_n b^2 (b\gamma)^{n-2} - a^2\Phi''(\chi)\sum_{n=1}^{\infty} A_n (b\gamma)^n + \\ 2(\chi - \chi_0)\Phi'(\chi)\sum_{n=1}^{\infty} nA_n b^2 (b\gamma)^{n-2} + \Phi(\chi)\sum_{n=1}^{\infty} nA_n b^2 (b\gamma)^{n-2} \\ + b^2\Phi(\chi)(1 - f(\chi))\sum_{n=1}^{\infty} A_n (b\gamma)^n = \frac{1}{2} f(\chi) J_0(b\gamma)$$

For $\chi = \chi_0$ the last equation becomes

$$A_1\Phi(\chi_0)\frac{1}{b\gamma} + 4A_2\Phi(\chi_0) + \sum_{n=1}^{\infty}[(n+2)^2\Phi(\chi_0)A_{n+2} \\ + (-a\Phi''(\chi_0) + (1 - f(\chi_0))\Phi(\chi_0)A_n]\gamma^n \\ = \frac{1}{2a} f(\chi_0)[1 - \frac{\gamma^2}{2^2} + \frac{\gamma^4}{2^2 4^2} - \frac{\gamma^6}{2^2 4^2 6^2} + ...]$$

and, therefore,

$$A_n = 0 \mathit{\ for\ odd\ } n,$$

$$A_2 = \frac{1}{2a}\frac{f(\xi_0)}{\varphi(\xi_0)}\frac{1}{2^2},$$

$$A_{2k+2} = [\frac{f(\xi_0)}{2a}\frac{(-1)^k}{2^2 4^2 ...(2k)^2} + (\frac{a^2}{b^2}\varphi''(\xi_0) - \\ (1 - f(\xi_0))\varphi(\xi_0))A_{2k}]\frac{1}{(2k+2)^2\varphi(\xi_0)},$$

$$k = 1, 2, 3...$$

Assume

$$\frac{a^2}{b^2}\Phi''(\chi_0) = (1 - f(\chi_0))\Phi(\chi_0), \Phi(\chi_0) = f(\chi_0) \qquad (17)$$

It follows that

$$A_{2k+2} = \frac{1}{2a}\frac{(-1)^k}{2^2 4^2 ...(2k)^2}, k = 1, 2, 3...$$

and

$$\omega(\chi, \gamma) = \frac{1}{2a}[J_0(b\gamma) + (1 - J_0(b\gamma))\varphi(\chi)] \qquad (18)$$

Substitute (18) into (16) and we obtain that the solution $\varphi(\chi)$ satisfies the following equation

$$-a(1 - J_0(b\gamma))\varphi''(\chi) + \\ (J_0(b\gamma) + J_2(b\gamma))(\chi - \chi_0)\varphi'(\chi) + \qquad (19) \\ (1 - f(\chi)(1 - J_0(b\gamma)))\varphi(\chi) = J_0(b\gamma) f(\chi)$$

and subject to the boundary conditions

$$\varphi(\chi_0) = f(\chi_0), \quad f(\pm\infty) = 0 \qquad (20)$$

## IV. Numerical Algorithm and Calculations

A solution $\omega(\chi_1, \tau_1)$ of the partial differential equation (15) at any particular point $(\chi_1, \tau_1)$ can be reduced to solving the boundary value problem (19), (20) with fixed value of $\gamma$ , where

$$\gamma = \sqrt{(\tau_1 - \tau_0)^2 - \frac{1}{a^2}(\chi_1 - \chi_0)^2} .$$

A family of curves $\gamma = const$ define hyperbolae imbedded into the characteristic triangle (Fig.4). The solution surface (15) can be represented as a family of curves over the hyperbolae in the 3D space $(\chi, \tau, \omega)$ . The surface representing the Riemann function for $\chi_0 = 0$ is shown in Fig. 5 and was drawn using MAPLE.

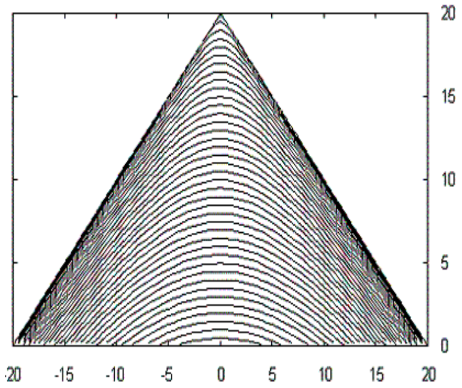


Fig. 4. The Characteristic triangle with the family of curves $\gamma = const$ .

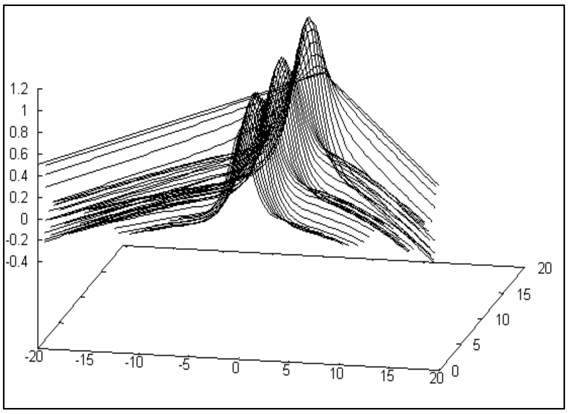


Fig. 5. The solution surface of (15) representing the Riemann function. Each curve is a solution of (19), (20) for particular fixed value of $\gamma$ .

The boundary value problem (19), (20) can be solved numerically using the relaxation method [9]. Since the solution of this BVP approaches zero when $\chi \to \pm\infty$ then for numerical calculations this problem can be solved for finite values of $\chi_{\lim}$ which are found manually from the condition $\lim_{\chi\to 0} f(\chi) \to 0$.

In the numerical calculations of the double integrals (13) and (14) one of the most difficult tasks is integration over the triangle $A$. First, the integral needs to be calculated for various values of $(\chi_0, \tau_0)$. The area of the rectangle A becomes larger when the value of $\tau_0$ increases. The second difficulty is that changing the value of the parameter $a$ leads to changing the rectangle $A$ shape. Therefore, it is very difficult to develop a universal algorithm for various values of $(\chi_0, \tau_0)$ and the parameter $a$ . The integration area $A$ (Fig. 3) can be mapped into a rectangle $R = \left\{ \frac{-1+V}{1+V} \le v \le 1, 0 \le u \le 1 \right\}$ (Fig. 6) with coordinates ($v$, $u$) by means of transformation

$$\chi = \chi_0 + a \cdot t_0 \left(1-u\right) v / \left(1-V\right),$$

$$\tau = t_0 u - V \chi_0 / a - V t_0 \left(1-u\right) v / \left(1-V\right),$$

where $t_0 = \tau_0 + V\chi_0 / a$ .

This transformation allows simplifying significantly numerical integration and speed up the algorithm of calculation of functions $u_{1,2}.(\chi, t)$. The area of integration can be covered by a rectangular mesh and then, the standard Simpson method can be used [9].

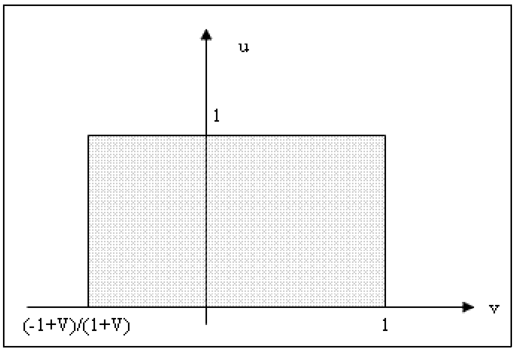


Fig. 6. The rectangular integration area R($v$, $u$).

The first integral (13) in the new variables ($v$, $u$) has the form

$$u_1\left(\chi_0, \tau_0\right) = \frac{t_0^2}{2}\left(1+V\right)\iint_R \{(1-u)\Phi(t_0 u)\times$$

$$\left[J_0\left(\alpha(v,u)\right) + \left\{1 - J_0\left(\alpha(v,u)\right)\right\}\varphi(v,u)\right]\}dudv \qquad (21)$$

$$\alpha(v,u) = t_0(1-u)\sqrt{(1+V)(1-v)(1+v+Vv-V})$$

where the double integral is calculated using the Simpson method and the value of $\varphi(v,u)$ at each point of the integral sum is a solution of the boundary value problem (19), (20).

The numerical algorithm was implemented in a program written in C++. For calculating values of the Bessel functions, double integrals and solving ordinary differential equations the code given in [9] was used.

As an example, the function $\Phi(t) = \sin(10t)$ was considered. Two graphs in a plane $(x, u_1)$ representing results of calculations of the integral (21) for $-1 \le x \le 1$ are shown in Fig. 7 and Fig. 8.

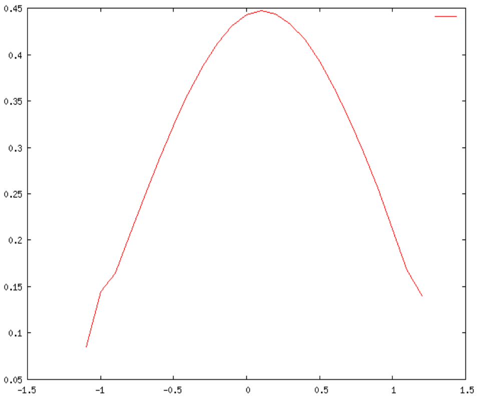


Fig. 7. $V = 0.95, t = \pi$

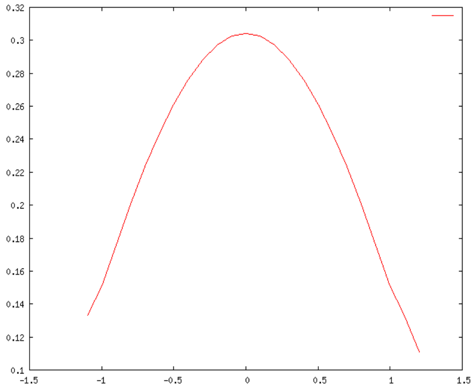


Fig. 8. $V = 0, t = 4088$

The values of $\varepsilon$ in (3) depend on a particular physical problem and are not discussed in this paper. They can be up to 0.3 in some cases.

## References

[1] E. Zauderer, “Partial differential equations of applied mathematics”, John Wiley & Sons Inc., USA, 1989.

[2] Ablowitz MJ, Segur H. “Solitons and the Inverse Scattering Transform”, SIAM, Philadelphia, 1981.

[3] J. B. Walsh, “An Introduction to stochastic partial differential equations”, Lecture Notes in Mathematics, 1180, Springer, pp. 266-437, 1986.

[4] B. Cairoli, J.B. Walsh, “Stochastic integrals in the plane”, Acta Matematica, 134, pp.111-183, 1975.

[5] R. Carmona, D. Nualart, “Random non-linear wave equations: Smothness of the solutions”, Prob. Theory Rel Fields, 79, pp.469-508, 1988.

[6] A. Bezen, “The Riemann’s function for a linear hyperbolic PDE”, Analysis paper, Department of Statistics, University of Melbourne, Report No 10, 1996.

[7] A. Bezen, F. Klebaner, “The Riemann’s function and its application to stochastic perturbations of a non-linear wave equation”, Random & Computational Dynamics, 5(4), pp.307-318, 1997.

[8] A. Bezen, Y. Stepanyants, “Kink propagation within the forced sine-Gordon equation”, Proceedings of III International Conference “Frontiers of Nonlinear Physics”, Nizhny Novgorod, 3-9 July, pp.50-51, 2007.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, “Numerical Recipes in C”, Cambridge Universiy Press, 1992.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 263–268

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Merging Jacobi and Gauss-Seidel Methods for Solving Markov Chains on Computer Clusters

Jarosław Bylina
Institute of Mathematics
Marie Curie-Sklodowska University
plac Marii Curie-Skłodowskiej 5, 20-031 Lublin, Poland
Email: jmbylina@hektor.umcs.lublin.pl

Beata Bylina
Institute of Mathematics
Marie Curie-Sklodowska University
plac Marii Curie-Skłodowskiej 5, 20-031 Lublin, Poland
Email: beatas@hektor.umcs.lublin.pl

***Abstract*—The authors consider the use of the parallel iterative methods for solving large sparse linear equation systems resulting from Markov chains—on a computer cluster. A combination of Jacobi and Gauss-Seidel iterative methods is examined in a parallel version. Some results of experiments for sparse systems with over $3 \times 10^7$ equations and about $2 \times 10^8$ nonzeros which we obtained from a Markovian model of a congestion control mechanism are reported.**

## I. Introduction and Motivation

DISCRETE-STATE models are widely employed for modeling and analysis of communication networks and computer systems. It is often convenient to model such a system as a continuous time Markov chain, provided probability distrbutions are assumed to be exponential (or combinations of exponential ones).

A CTMC (Coninuous-Time Markov Chain) may be represented by a set of states and a transition rate matrix $\mathbf{Q}$ containing state transition rates as coefficients, and can be analysed by using probabilistic model checking. Such an analysis proceeds by specifying desired performance properties as some temporal logic formulae, and by verifying these properties automatically, using the appropriate model checking algorithms. A core component of these algorithms is a computation of the steady-state probabilities of the CTMC. This is reducible to the classical problem of solving a (homogeneous) sparse system of linear equations, of the form $\mathbf{Ax} = \mathbf{b}$, of size equal to the number of states in the CTMC.

A limitation of the Markovian modeling approach is the fact that the CTMC models tend to grow extremely large due to the state space explosion problem. This is caused by the fact that a system is usually composed of a number of concurrent subsystems, and that the size of the state space of the overall system is generally exponential in the number of subsystems. A realistic system can give rise to a large state space, typically over $10^6$. As a consequence, much research is focused on the development of techniques, that is, methods and data structures, which minimise the computational (space and time) requirements for analysing large and complex systems.

One of such techniques is parallelization. Problems of parallel computations for such systems and finding its steady-state probabilities in parallel is brought up in [3], [8], [13], [15]. In [2], [4], [7], [17] a manner of distributed generation of the matrix $\mathbf{Q}$ in a network environment is desribed and in [6] a similar algorithm—on a computer cluster—is presented. Paper [5] describes solving sparse linear systems with the use of the GMRES algorithm [18] in a network environment. The parallel Jacobi method was discussed and a parallel method for the CTMC steady-state solution is presented in [15]. The Gauss-Seidel method is used for parallel solving of Markov chains in [14], [20].

In this paper a combination of two classical iterative methods for solving linear equation systems, namely Jacobi method and Gauss-Seidel method is presented. These methods were chosen because the presented algorithm is intended for computer clusters and Jacobi method is inherently parallel (Gauss-Seidel method has not got such a property and its parallelization requires a lot of communication), but Gauss-Seidel method usually converges much faster than Jacobi method [1]. Properties of such a combined method are experimantally examined in this paper. We try to study relative speedup and efficiency of the algorithm—as the traditional characteristics of parallel algorithms.

The rest of the paper is organized as follows. Section II presents the problem. In Section III classical iterative methods are reminded. Section IV shows used data distribution. Section V presents traditional block/parallel iterative methods and (in Section V-C) an approach employed by the authors. Section VI describes conducted experiments and Section VII contains some conclusions.

## II. CTMCs and the Steady-State Solution

While modeling with Markov chains, in a steady state (independent of time), we obtain a linear equation system like follows;

$$\mathbf{Q}^T\mathbf{x} = \mathbf{0}, \qquad \mathbf{x} \geq \mathbf{0}, \qquad \mathbf{x}^T\mathbf{e} = 1 \tag{1}$$

where $\mathbf{Q}$ is a transition rate matrix, $\mathbf{x}$ is an unknown vector of states probabilities and $\mathbf{e} = (1, 1, ...., 1)^T$. The matrix $\mathbf{Q}$

This work was partially supported within the project *Metody i modele dla kontroli zatłoczenia i oceny efektywności mechanizmów jakości usług w Internecie następnej generacji* (N517 025 31/2997). This work was also partially supported by Marie Curie-Sklodowska University in Lublin within the project *Równoległe algorytmy generacji i rozwiązywania mechanizmów kontroli przeciążenia w protokole TCP modelowanych przy użyciu łańcuchów Markowa*.

is a square one of size $n \times n$, usually a big one, of rank $n-1$, sparse, with dominant diagonal. It is also a singular matrix demanding adequate methods to solve the equation. Markovian models solving demands overcoming both numerical and algorithmic problems. Solving the equation system (1) generally requires applying iterative methods, projection methods or decomposition methods but occasionally (for the need of an accurate solution) direct methods are used as well. The rich material concerning the methods mentioned above can be found in [19].

## III. Iterative methods—Jacobi and Gauss-Seidel

In this section classical iterative methods are reminded. The general form of such a method step is

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{M}^{-1}\mathbf{N}\mathbf{x}^{(k)} \tag{2}$$

where

$$\mathbf{M} - \mathbf{N} = \mathbf{Q}^T. \tag{3}$$

### A. The Method of Jacobi

The method of Jacobi is a classical iterative method with the coefficient matrix $\mathbf{Q}^T$ split as following:

$$\mathbf{Q}^T = \mathbf{D} - (\mathbf{L} + \mathbf{U}) \tag{4}$$

which corresponds to assigning:

$$\mathbf{M} = \mathbf{D}, \qquad \mathbf{N} = \mathbf{L} + \mathbf{U} \tag{5}$$

in (3). The matrix $\mathbf{D}$ is a diagonal matrix, the matrix $\mathbf{L}$ is a strictly lower triangular matrix (with zeroes on its diagonal) and the matrix $\mathbf{U}$ is a strictly upper triangular matrix (with zeroes on its diagonal). So in this method the step (2) looks as following:

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} \tag{6}$$

and in scalar form (for $i = 1, \ldots, n$):

$$x_i^{(k+1)} \leftarrow \frac{1}{d_{ii}} \left( \sum_{j=1}^{i-1} l_{ij} x_j^{(k)} + \sum_{j=i+1}^{n} u_{ij} x_j^{(k)} \right). \tag{7}$$

The method of Jacobi is very convinient to vectorize and to parallelize. It can also be seen in the Jacobi equation (7) that the new approximation of the solution vector ($\mathbf{x}^{(k+1)}$) is calculated by using only the old approximation of the vector ($\mathbf{x}^{(k)}$). This method, therefore, possess a high degree of natural parallelism. However, Jacobi methods has relatively slow convergence.

### B. The Method of Gauss-Seidel

In this method we have the same splitting of the matrix $\mathbf{Q}^T$ but with a different grouping of components:

$$\mathbf{Q}^T = (\mathbf{D} - \mathbf{L}) - \mathbf{U} \tag{8}$$

(the matrices $\mathbf{D}$, $\mathbf{L}$, $\mathbf{U}$ are defined as in the previous section). Here we have $\mathbf{M} = (\mathbf{D} - \mathbf{L})$ and $\mathbf{N} = \mathbf{U}$, so the step (2) is:

$$\mathbf{x}^{(k+1)} \leftarrow (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}\mathbf{x}^{(k)} \tag{9}$$

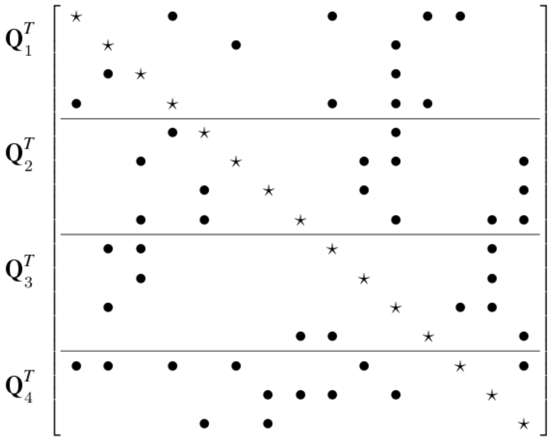


Fig. 1. An example of the matrix $\mathbf{Q}^T$ division

and in scalar form (for $i = 1, \ldots, n$):

$$x_i^{(k+1)} \leftarrow \frac{1}{d_{ii}} \left( \sum_{j=1}^{i-1} l_{ij} x_j^{(k+1)} + \sum_{j=i+1}^{n} u_{ij} x_j^{(k)} \right). \tag{10}$$

The method of Gauss-Seidel is not so convinient for vectorization or parallelization, but it converges faster—what can be explained by the fact that although both methods use the same scalar formulas (7) and (10), but in the method of Gauss-Seidel the newly computed approximation of a solution component is used as soon as it is available.

The Gauss-Seidel method typically converges faster than the Jacobi method by using the most recently available approximations of the elements of the iteration vector. The other advantage of the Gauuss-Seidel algorithm is that it can be implemented using only one iteration vector, which is important for large linear equation systems where storage of single iteration vector alone may require 10GB or more. However, a consequence of using the most recently available solution approximation is that the method is inherently sequential—it does not possess natural parallelism.

## IV. Data Distribution

In parallel programming a very crucial issue is a division of data among computational nodes (machines, processors etc.). In the algorithm described later (in Section V-C) the matrix $\mathbf{Q}^T$ is divided among cluster nodes as in Figure 1—that is the matrix is divided into $p$ rectangular submatrices $\mathbf{Q}_i^T$, each stored in the $i$th cluster node.

Each submatrix is a sparse matrix and takes part in the computations on its node (where it is stored).

## V. A Parallel Method of Jacobi/Gauss-Seidel

The description of the parallel Jacobi/Gauss-Seidel method will be started from the presentation of the well-known block iterative methods, namely block Jacobi method and block Gauss-Seidel method.

Block-based formulations of the iterative methods which perform matrix computations on block-by-block basis usually turn out to be more efficient and easier to parallelize. Generally iterative block methods demand more calculations per iteration, which is recompensed by a faster convergence rate (and sometimes better cache utilization).

It is possible in Markov chains to divide the transition rate matrix into blocks (or even generate it in blocks straight away [4], [7]) and develop iterative methods basing on that division.

In Markov chains problems it is often the case that the state space can be meaningfully partitioned into subsets and thus, it is possible to partition the transition rate matrix respectively and to base the solution method on blocks implied by such a partition.

The homogeneous equations system (1) is divided into $K^2$ square blocks of the same size in the following way:

$$\begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \dots & \mathbf{Q}_{1K} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} & \dots & \mathbf{Q}_{2K} \\ \dots & \dots & \dots & \dots \\ \mathbf{Q}_{K1} & \mathbf{Q}_{K2} & \dots & \mathbf{Q}_{KK} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} = \mathbf{0}. \tag{11}$$

We introduce block splitting:

$$\mathbf{Q}^T = \mathbf{D}_K - (\mathbf{L}_K + \mathbf{U}_K), \tag{12}$$

where $\mathbf{D}_K$ is a block-diagonal matrix, $\mathbf{L}_K$ is a strictly block lower triangular matrix, $\mathbf{U}_K$ is a strictly block upper triangular matrix with form:

$$\mathbf{D}_K = \begin{bmatrix} \mathbf{D}_{11} & 0 & \dots & 0 \\ 0 & \mathbf{D}_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{D}_{KK} \end{bmatrix}, \tag{13}$$

$$\mathbf{L}_K = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \mathbf{L}_{21} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \mathbf{L}_{K1} & \mathbf{L}_{K2} & \dots & 0 \end{bmatrix}, \tag{14}$$

$$\mathbf{U}_K = \begin{bmatrix} 0 & \mathbf{U}_{12} & \dots & \mathbf{U}_{1K} \\ 0 & 0 & \dots & \mathbf{U}_{2K} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \tag{15}$$

where $\mathbf{D}_{ii} = \mathbf{Q}_{ii}$, $\mathbf{L}_{ij} = -\mathbf{Q}_{ij}$, $\mathbf{U}_{ij} = -\mathbf{Q}_{ij}$.

### A. A Block Jacobi Algorithm

The iterative method of Jacobi was described in Section III-A. In this section, a block parallel Jacobi algorithm for the solution of the linear equation system (1) is presented.

Block Jacobi method is given by (for $i = 1, \dots, K$):

$$\mathbf{Q}_{ii}^T \mathbf{x}_i^{(k+1)} = -\sum_{j<>i} \mathbf{Q}_{ij}^T \mathbf{x}_j^{(k)} \tag{16}$$

where blocks are as in (11).

Having $K$ computational nodes (computers or processors), every equation of (16) can be solved independently by a computational node. The equtions (16) can be solved by an arbitrary method.

### B. A Block Gauss-Seidel Algorithm

Just like the scalar Gauss-Seidel algorithm (see Section III-B), the block Gauss-Seidel algorithm can be written:

$$(\mathbf{D}_K - \mathbf{L}_K)x^{(i+1)} = \mathbf{U}_K x^{(i)}. \tag{17}$$

Describing the equation mentioned above in a scalar-like form we get (for $j = 1, \dots, K$):

$$\mathbf{Q}_{jj}\mathbf{x}_j^{(i+1)} = -\left( \sum_{l=1}^{i-1} \mathbf{Q}_{jl}\mathbf{x}_l^{(i+1)} + \sum_{l=j+1}^{K} \mathbf{Q}_{jl}\mathbf{x}_l^{(i)} \right) \tag{18}$$

where $\mathbf{x}_j^{(i)}$ is the $j$th $(n/K)$-element subvector of the vector $\mathbf{x}^{(i)}$ (as in (11)).

As a result of above in every step we must solve $K$ equation systems of $n/K$ size each in the following form (for $j = 1, \dots, K$):

$$\mathbf{Q}_{jj}\mathbf{x}_i^{(i+1)} = \mathbf{z}_j^{(i+1)} \tag{19}$$

where

$$\mathbf{z}_j^{(i+1)} = -\left( \sum_{l=1}^{i-1} \mathbf{Q}_{jl}\mathbf{x}_l^{(i+1)} + \sum_{l=j+1}^{K} \mathbf{Q}_{jl}\mathbf{x}_l^{(i)} \right). \tag{20}$$

We can apply different direct and iterative methods to the solve equation (19). There is a small number of iterations demanded to obtain convergence for a small number of blocks (i.e. submatrices are big). The difficulty is that it is very hard to parallelize effectively—it is caused by the use of the newly computed values just after their computation.

### C. A Modified Block Jacobi Algorithm

In this section we present an algorithm, which takes advantage of the division of the matrix $\mathbf{Q}$ between computational nodes described in Section IV. The algorithm proposed here is a combination of Jacobi and Gauss-Seidel iterative methods.

To start from a usual Jacobi method, there is

$$\mathbf{Q}^T = \mathbf{D} - (\mathbf{L} + \mathbf{U}) \tag{21}$$

(see Section III-A).

Let

$$\mathbf{H} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}). \tag{22}$$

Thus, the (6) can be written

$$\mathbf{x}^{(k+1)} = \mathbf{H}\mathbf{x}^{(k)}. \tag{23}$$

In a block iterative method the linear system (23) is divided into some subsystems. In this section the matrix $\mathbf{H}$ is divided into $p$ blocks ($p$ is the number of computatinal nodes), each block $\mathbf{H}_i$ ($i = 1, \dots, p$) of the size of $n$ columns and $n/p$ rows (last one, $\mathbf{H}_p$ can be shorter—as in Figure 1—it does not influences general considerations). Such a division corresponds to the division of the matrix $\mathbf{Q}^T$ proposed in Section IV, because to obtain elements of the matrix $\mathbf{H}_i$ one needs only elements of the matrix $\mathbf{Q}_i^T$.

Similarly, the vector $\mathbf{x}$ (and some auxiliary vectors in the implementation) is divided into $p$ subvectors, each of the size $n/p$.

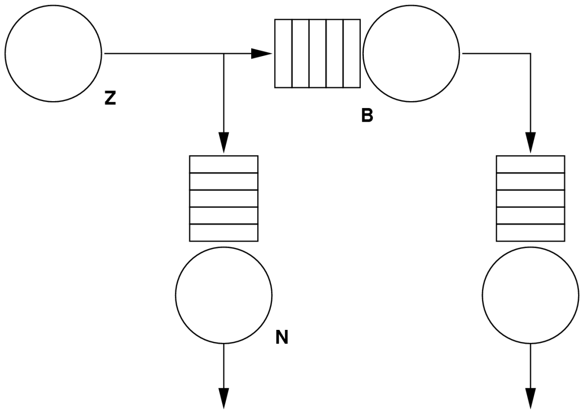


Fig. 2. A Markovian queuing model of the tail-drop mechanism

Now, the (23) can be written:

$$\begin{bmatrix} \mathbf{x}_1^{(k+1)} \\ \mathbf{x}_2^{(k+1)} \\ \vdots \\ \mathbf{x}_p^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_p \end{bmatrix} \mathbf{x}^{(k)} \tag{24}$$

and, in another manner, for $i = 1, \ldots, p$:

$$\mathbf{x}_i^{(k+1)} = \mathbf{H}_i \mathbf{x}^{(k)}. \tag{25}$$

(25) is a formula for the block Jacobi method (Section III-A). All the equations in (25) are solved independently, so it is very suitable to parallelize for $p$ processors – each of them solves one equation and then they exchange the resulting vectors $\mathbf{x}_i^{(k+1)}$ to build its new version $\mathbf{x}^{(k+1)}$.

However, the Jacobi method is rather slowly convergent, so in the presented algorithm, every computational node can employ the formula of the Gauss-Seidel algorithm (10) instead of the Jacobi algorithm (7)—using the newly obtained elements of the vector $\mathbf{x}_i^{(k+1)}$ (although only those which are stored in the same computational nodes) and hoping for the better convergency.

In other words, the algorithm can be described as a block Jacobi iterative method with solving inner blocks with the Gauss-Seidel iterative method.

In borderline cases the presented algorithm reduces to the pure Jacobi algorithm (for $p = n$) and to the pure Gauss-Seidel algorithm (for $p = 1$).

### D. Implementation details

We propose an algorithm for homogeneous cluster environments. This algorithm is based on a message-passing paradigm and consists of one module for each of nodes. The algorithm presented below is just a skeleton, and the detailed implementation, such as data preparation, parameters passing, and so forth, might be different according to requirements of various applications.

The algorithm is composed of several steps. First, the starting information is acquired. Next, the matrix $\mathbf{Q}$ is generated in parts so that every node keeps only needed states (as in Figure 1) Next, in loop, in every node for its block we make a step of the Gauss-Seidel method.

The algorithm for every node (in pseudo-code) is described as follows:

```
Initialization(&MyNumber);

FindBeginAndEnd(MyNumber, &n0, &n1);
  /* computes starting index n0 and
     ending index n1 for the block of
     the node MyNumber */
QT=Generate(n0, n1);
  /* generates adequate block */
FillVector(X, 1.0/n);
IterNo=0;

do
{
  IterNo++;
  InnerGaussSeidel(QT, X);
    /* the node computes only its own
       part of the vector */
  GatherVector(X)
    /* every node needes the whole vector
       for further computations */
  Normalize(X);
} while(Remainder(QT, X)>EPSILON &&
        IterNo<MAXITER);

Finalizetion();
```

## VI. Experimental Results

### A. Properties of the Used Matrices

The matrices obtained for tests were transition rate matrices for a very simple model of a tail-drop mechanism (a very simple congestion control mechanism in a router's buffer). The queuing model for such a mechanism is shown in Figure 2. It is a passive mechanism, that is, it does not make any decision, as far as the moment of dropping packets is concerned—they are always dropped when there is no room for new packets. Moreover, in the tail-drop mechanism the packet that just arrived, is dropped. A Markovian queuing model for such a mechanism consists of: a source **Z** (which sends packets with variable rates), a service station **B** (which corresponds to a router's buffer) and two auxiliary service stations **A** and **N** (which correspond to confirmations and rejections, respectively, returning to the source **Z**).

The rate of the source **Z** is not constant but it increases (not above a given maximum) as confirmations leave the station **A** and it decreases (not below a given minimum) as rejections leave the station **N** (both events represent reaching the source by the information about the packet's fate). The states of such a model can be written as vectors of numbers. In our example it would be $(l, b, a, m)$, where $l$ is an integer between 1 and $l_{max}$ and it means the current relative intensity of the source **Z** and $b$, $a$ and $m$ are integers between 0 and $b_{max}$, $a_{max}$,

TABLE I
TEST MATRICES AND THE ALGORITHM PERFORMANCE

| $l_{max}$ | $b_{max}$ | $a_{max}$ | $m_{max}$ | $n$ | $p$ | $T_p$ [s] | $S_p$ | $E_p$ |
|---|---|---|---|---|---|---|---|---|
| 40 | 40 | 40 | 40 | 2 756 840 | 1 | 145 | — | — |
| | | | | | 10 | 96 | 1.51 | 0.15 |
| 50 | 50 | 50 | 50 | 6 632 550 | 1 | 480 | — | — |
| | | | | | 10 | 210 | 2.29 | 0.23 |
| 60 | 60 | 60 | 60 | 13 618 860 | 1 | 812 | — | — |
| | | | | | 10 | 399 | 2.04 | 0.20 |
| 70 | 70 | 70 | 70 | 25 053 770 | 1 | 3 373 | — | — |
| | | | | | 10 | 714 | 4.72 | 0.47 |
| 72 | 72 | 72 | 72 | 28 009 224 | 1 | 4 283 | — | — |
| | | | | | 10 | 829 | 5.17 | 0.52 |
| 74 | 74 | 74 | 74 | 31 218 750 | 1 | 5 232 | — | — |
| | | | | | 10 | 929 | 5.63 | 0.56 |

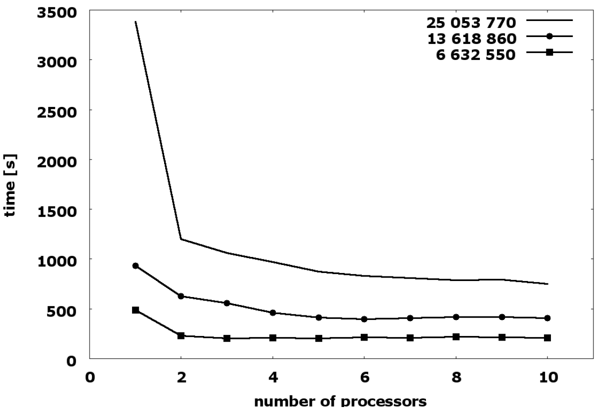


Fig. 3. Performance time of the algorithm for three different sizes of matrices

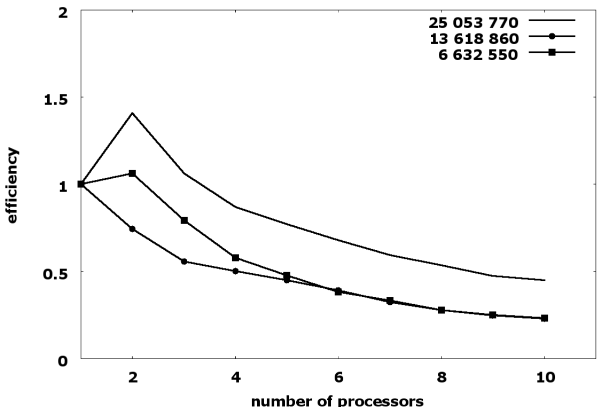


Fig. 5. Efficiency of the algorithm for three different sizes of matrices

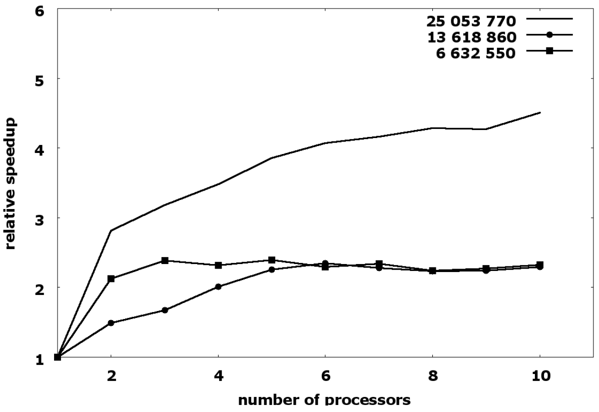


Fig. 4. Relative speedup of the algorithm for three different sizes of matrices

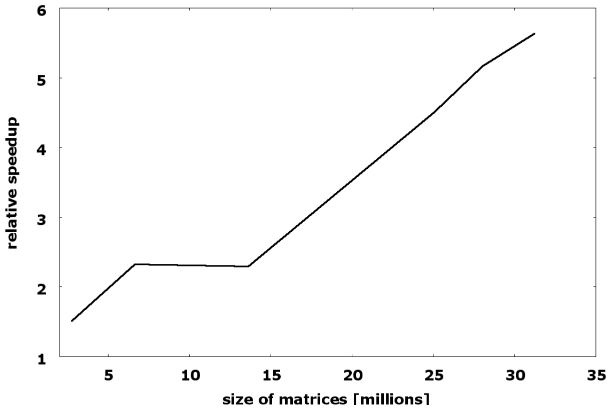


Fig. 6. Relative speedup of the algorithm working on 10 processors for various matrices

$m_{max}$ (respectively) and they mean the number of packets waiting or being processed currently in the stations **B**, **A**, **N** (respectively).

### *B. Computational Environment*

Experiments were carried out in a cluster environment (part of CLUSTERIX – a cluster of Linux machines consisting of more than 800 machines distributed all over Poland and connected with a fast optical network; one of the fastest European distributed supercomputers [21]) dedicated for computing.

The CLUSTERIX itself is built as a cluster of local clusters. So up to 10 machines were used—all belonging to a single local cluster— as we were interested in investigating behavior of the algorithm in such an environment. Each of the machines was equipped with two 64-bit processors Intel Itanium 2 1.4 GHz and 4 GB RAM, but we used only one processor in every machine, because we were interested in distributed results. The cluster nodes were used when they were idle—in order to assure credible performance times.

The algorithm was implemented with the use of the MPI (message passing interface) standard and MPICH library.

### C. Numerical Results

The properties of the models and the matrices used in tests are shown in Table I. All the matrices have about 5–6 elements in a row (very sparse matrices). There are also some characteristics of the performance—$T_p$, $S_p$ and $E_p$ presented there. The distributed run-time of 50 iterations of the algorithm is measured as the maximum run-time from the start and we denote it as $T_p$ (for $p$ machines). The speedup for $p$ processors is denoted by $S_p$ and is given by $S_p = T_1/T_p$. The efficiency for $p$ processors is denoted by $E_p$ and is defined as $E_p = S_p/p$.

We can see in Figures 3–6 and in Table I that the relative speedup and the efficiency grows with the growth of the size of matrices, so we can expect that for bigger matrices we can get better results.

## VII. Conclusion

The authors have developed a parallel and somewhat scalable algorithm that computes the vector $\mathbf{x}$ from the equation (1) for a very large $n$; the matrix $\mathbf{Q}$ is distributed among the computational nodes. To find $\mathbf{x}$, a parallel combination of Jacobi and Gauss-Seidel iterative methods was used. The numerical experiments on a parallel system have been carried out in order to assess the effectiveness of the distributed algorithm on a computer cluster. The numerical tests indicate that some efficiency is possible if the sufficient amount of work per processor is provided (for small sizes of the matrix the scalability is worse). The parallel implementation was benchmarked using a Markovian model of congestion control.

The results suggest an important area for future research—writing numerical algorithms (which find probability vectors) for huge matrices modeling Markov chains distributed in row-striped manner among many processor (as in our algorithm) and performance optimization—perhaps using preconditioned iterative methods [2], [9], [10], [11], [12], [17].

The proposed method is tested on matries connected with Markov chains, but we can use this method to different application to other problems with sparse matrices.

In the future we want to examine the combination of Jacobi and Gauss-Seidel methods theoretically and compare our approach with others.

## References

[1] J. M. Bahi, S. Contessot-Vivier, R. Coutier: Parallel iterative algorithms. From sequential to Grid Computing. Chapman & Hall/CRC, 2007.

[2] M. Benzi, M. Tuma: A parallel solver for large-scale Markov chains. *Applied Numerical Mathematics,* 41 (2002), pp. 135–153.

[3] P. Buchholz, M. Fischer, P. Kemper: Distrbuted steady state analysis using Kronecker algebra. *Proceedings of the Third International Conference on the Numerical Solution of Markov Chains (NSNC '99),* Zaragoza, Spain, September 1999, pp. 76–95.

[4] J. Bylina: A distributed approach to solve large Markov chains. *Proceedings from EuroNGI Workshop: New Trends in Modeling, Quantitive Methods and Measurements,* Jacek Skalmierski Computer Studio, Gliwice 2004, pp. 145–154.

[5] J. Bylina, B. Bylina: A Markovian model of the RED mechanism solved with a cluster of computers. *Annales UMCS Informatica,* 5 (2006), pp. 19–27.

[6] J. Bylina, B. Bylina: Analysis of a Parallel Algorithm of Distributed of Matrix for Markovian Models of Congestion Control. Wydawnictwa Komunikacji i Łączności, Warszawa 2008,

[7] J. Bylina, B. Bylina: Distributed generation of Markov chains infinitesimal generators with the use of the low level network interface. *Proceedings of 4th International Conference Aplimat 2005*, part II, pp. 257–262.

[8] N. J. Dingle, P. G. Harrison, W. J. Knottenbelt: Uniformization and hypergraph partitioning for the distributed computation of response time densities in very large Markov models. *Journal of Parallel and Distributed Computing,* 64 (2004) pp. 908–920.

[9] K. M. Giannoutakis, G. A. Gravvanis, B. Clayton, A. Patil, T. Enright, J. P. Morrison: Matching high performance approximate inverse preconditioning to architectural platforms, *The Journal of Supercomputing,* 42(2), 2007, pp. 145–163.

[10] G. A. Gravvanis: Explicit Approximate Inverse Preconditioning Techniques, *Archives of Computational Methods in Engineering,* 9(4), 2002, pp. 371–402.

[11] G. A. Gravvanis: Explicit preconditioned generalized domain decomposition methods, *International Journal of Applied Mathematics,* 4(1), 2000, pp. 57–71.

[12] G. A. Gravvanis, V. N. Epitropou, K. M. Giannoutakis: On the performance of parallel approximate inverse preconditioning using Java multithreading techniques, *Applied Mathematics and Computation,* Vol. 190, 2007, pp. 255–270.

[13] W. Knottenbelt: Distributed disk-based solution techniques for large Markov models. *Proceedings of the Third International Conference on the Numerical Solution of Markov Chains (NSNC '99)*, Zaragoza, Spain, September 1999.

[14] M. Kwiatkowska, D. Parker, Y. Zhang, R. Mehmood: Dual-processor parallelisation of symbolic probabilistic model checking. In *Proceedings of 12th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04),* 2004, pp. 123–130.

[15] R. Mehmood, J. Crowcroft: Parallel Iterative Solution Method for Large Sparse Linear Equation Systems Technical Report UCAM-CL-TR-650, Computer Laboratory, University of Cambridge, UK October 2005.

[16] Ng Chee Hock: Queuing Modelling Fundamentals. Wiley, New York, 1996.

[17] A. Platis, G. Gravvanis, K. Giannoutakis, E. Lipitakis: A two-phase cyclic non-homogeneous markov chain performability evaluation by explicit approximate inverses applied to a replicated database system, *Journal of Mathematical Modelling and Algorithms,* 2, 2003, pp. 235–249.

[18] Y. Saad, M. H. Schultz: GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM Journal of Scientific and Statistical Computing,* 7, 1986, pp. 856–869.

[19] W. Stewart: Introduction to the Numerical Solution of Markov Chains, Princeton University Press, Chichester, West Sussex 1994.

[20] Y. Zhang, D. Parker, M. Kwiatkoska: A wavefront parallelisation of CTMC solution using MTBDDs. In *Proceedings of International Conference on Dependable Systems and Networks (DSN'05),* IEEE Computer Society Presss, 2005, pp. 732–742.

[21] http://clusterix.pcz.pl/

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 269 – 275

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Influence of molecular models of water on computer simulations of water nanoflows

Janusz Bytnar
Institute of Technical Engineering,
PWSZ, Czarnieckiego 16
37-500 Jaroslaw
Poland

Anna Kucaba-Piętal
Rzeszow University of
Technology,
The Faculty of Mechanical
Engineering and Aerodynamics,
Powstanców Warszawy 8,
35-959 Rzeszów,
Institute of Technical Engineering,
PWSZ, Czarnieckiego 16
37-500 Jaroslaw
Poland

Zbigniew Walenta
Institute of Fundamenal
Technological Research PAN,
Department of Mechanics and
Physics of Fluids,
Swietokrzyska 21,
00-049 Warsaw,
Poland

***Abstract*—We present some problems related to the influence of molecular models on Molecular Dynamics simulation of water nanoflows. Out of large number of existing models of water we present some results of the MD simulation of water nanoflows for four molecular models: TIP4P, PPC TIP4P-2005, TIP5P.**

## I. Introduction

COMPUTER simulation methods have become a very powerful tool for solving many-body problems in statistical physics, physical chemistry and biophysics. Although theoretical description of complex systems in the framework of statistical physics is well developed and the experimental techniques for obtaining detailed microscopic information are quite sophisticated, it is often only possible to study specific aspects of those systems in detail via simulation. On the other hand, simulations need specific input parameters characterizing systems in question, which either come from theoretical considerations or are provided by experiment [1].

The deterministic method of Molecular Dynamics (MD) simulation [2], although theoretically valid for the whole range of densities, is employed mainly for liquids and solids. The long flight paths between collisions of gas molecules make the method of Molecular Dynamics prohibitively expensive, while other methods, like e.g. Direct Monte-Carlo Simulation, can give satisfactory results at much lower computational cost. In liquids the molecules are densely packed and remain in constant contact with the neighbours. Under such conditions Molecular Dynamics seems to be the most accurate and, at the same time, the most efficient simulation method.

Molecular Dynamics requires description of the molecules and the forces acting between them. Perhaps the most often, to describe the forces, the Lennard-Jones potential is used; it assumes that the  molecules are spherically symmetric, repelling one another at close and attracting at far distances.

The motivation for this research stems from real nanochannel flow problem. For flows of liquids in nanochannels the continuum description is no longer valid and the Molecular Dynamic Simulation seems to be the only appropriate approach to the problem. In the following some MD flow simulation results will be presented to illustrate usefulness of the method.

## II. Molecular models of water

The structure of the water molecule (Fig. 1) is relatively complex and can only be properly described in the framework of quantum mechanics. However, this kind of description is not applicable for Molecular Dynamics simulation, therefore a number of simplified models have been proposed. Unfortunately, each of them has only limited range of applications.

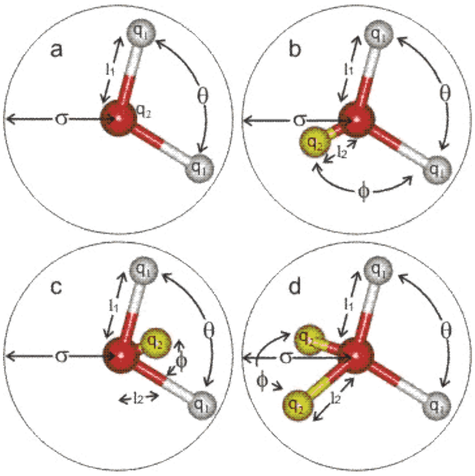


Fig.  1 Most frequently used geometrical model shapes of water molecule

In Table 1 and Fig. 2 we present four molecular models of water, selected for our Molecular Dynamics simulations of flow in nanochannels. Table 2 contains the values of some physical parameters of water, calculated with the use of those models, together with corresponding experimental data.  The molecular parameters of copper from [6] and [7] were used.

TABLE I.
PARAMETERS OF THE WATER MOLECULAR MODELS USED IN OUR SIMULATION [13]

| Model | Geometrical configuration (Fig. 2) | Lennard-Jones molecular diameter б , $1*10^{-10}$ m | Lennard-Jones potential well depth ε kJ/mol | $l_{1,}$ $1*10^{-10}$ m | $l_2$ $1*10^{-10}$ m | $q_1$ (e) | $q_2$ (e) | $\theta$ ° | $\Phi$ ° |
|---|---|---|---|---|---|---|---|---|---|
| TIP4P | c | 3.15365 | 0.6480 | 0.9572 | 0.15 | +0.5200 | -1.0400 | 104.52 | 52.26 |
| TIP4P/2005 | c | 3.1589 | 0.7749 | 0.9572 | 0.1546 | +0.5564 | -1.1128 | 104.52 | 52.26 |
| TIP5P | d | 3.12000 | 0.6694 | 0.9572 | 0.7 | +0.2410 | -0.2410 | 104.52 | 109.47 |
| PPC | b | 3.23400 | 0.6000 | 0.9430 | 0.06 | +0.5170 | -1.0340 | 106.00 | 127.00 |

TABLE II.
CALCULATED PHYSICAL PROPERTIES OF THE WATER MODELS USED IN OUR SIMULATION [13]

| Model | Dipole m oment μ = qi*li, m | Dielectric constant | Self diffusion $10^{-9}$ m2/s | Average configurational energy kJ/mol | Density maximum, °C | Expansion coefficient $10^{-4}$ °C$^{-1}$ |
|---|---|---|---|---|---|---|
| TIP4P | 2.18 | 53 | 3.29 | -41.8 | -25 | 4.4 |
| TIP4P/2005 | 2.305 | 60 | 2.08 | - | +5 | 2.8 |
| TIP5P | 2.29 | 81.5 | 2.62 | -41.3 | +4 | 6.3 |
| PPC | 2.52 | 77 | 2.6 | -43.2 | +4 | - |
| Experimental | 2.65 | 78.4 | 2.30 | -41.5 | +3.984 | 2.53 |

## III. MOLECULAR DYNAMICS

Molecular Dynamics simulation method treats the medium as an ensemble of molecules. Each molecule may consist of one or more atoms attached to each other in the way specific for given substance. It is assumed that:

- each atom is treated as a point mass,
- simple force rules describe the interactions between atoms; force acting on a molecule is a sum of forces acting on all constituent atoms,
- Newton's equations of motion are integrated to obtain coordinates and velocity of each molecule as a function of time (see Fig.2)
- thermodynamic statistics are extracted from positions and velocities of the molecules.

The orientation of the molecules can be represented in several ways, however the use of quaternions [4] seems to be the most advisable. The most important advantage of quaternions is the fact, that they lead to equations of motion free of singularities (which is not the case for e.g. Euler angles). This, in turn, leads to good numerical stability of the simulation.

Integration algorithms used in Molecular Dynamics simulation are based on finite difference methods, with discretized time and the time step equal to **Δt.** Knowing the positions and some of their time derivatives at time t (the exact details depend on the type of algorithm), the integration scheme gives the same quantities at a later time (**t** + **Δt**). With such procedure the evolution of the system can be followed for long times [3].

**Stages of simulation** :

- Initiation: placing the molecules of water and the copper atoms in the knots of crystalline mesh. After that the velocities of the molecules are initialized. Their values are sampled at random from the Maxwell – Boltzmann distribution for the assumed temperature.
- Balancing: after initiation the positions of molecules are far from equilibrium. The whole ensemble is allowed to move freely for some time to attain equilibrium positions. This is always connected with decreasing the potential and increasing the kinetic energy of the molecules, i.e. increasing the temperature of the medium. This excess temperature must be removed by a suitable "thermostat".
- Actual simulation: after attaining equilibrium, the simulation starts. The required data (specified in advance) are accumulated in "dump-files" in preselected time intervals. Any property of interest, dynamic or static, may then be evaluated with the use of the data in the dump.

In molecular dynamics we follow the laws of classical mechanics,

$$F_i = m_i * a_i$$

for each atom i in system constituted by N atoms. Here $m_i$ is the atom mass, $a_i = d^2 r_i / dt^2$ its acceleration, and $F_i$ the force acting upon it, due to the interactions with other atoms [3].

The motion is governed by the Newton-Euler equations

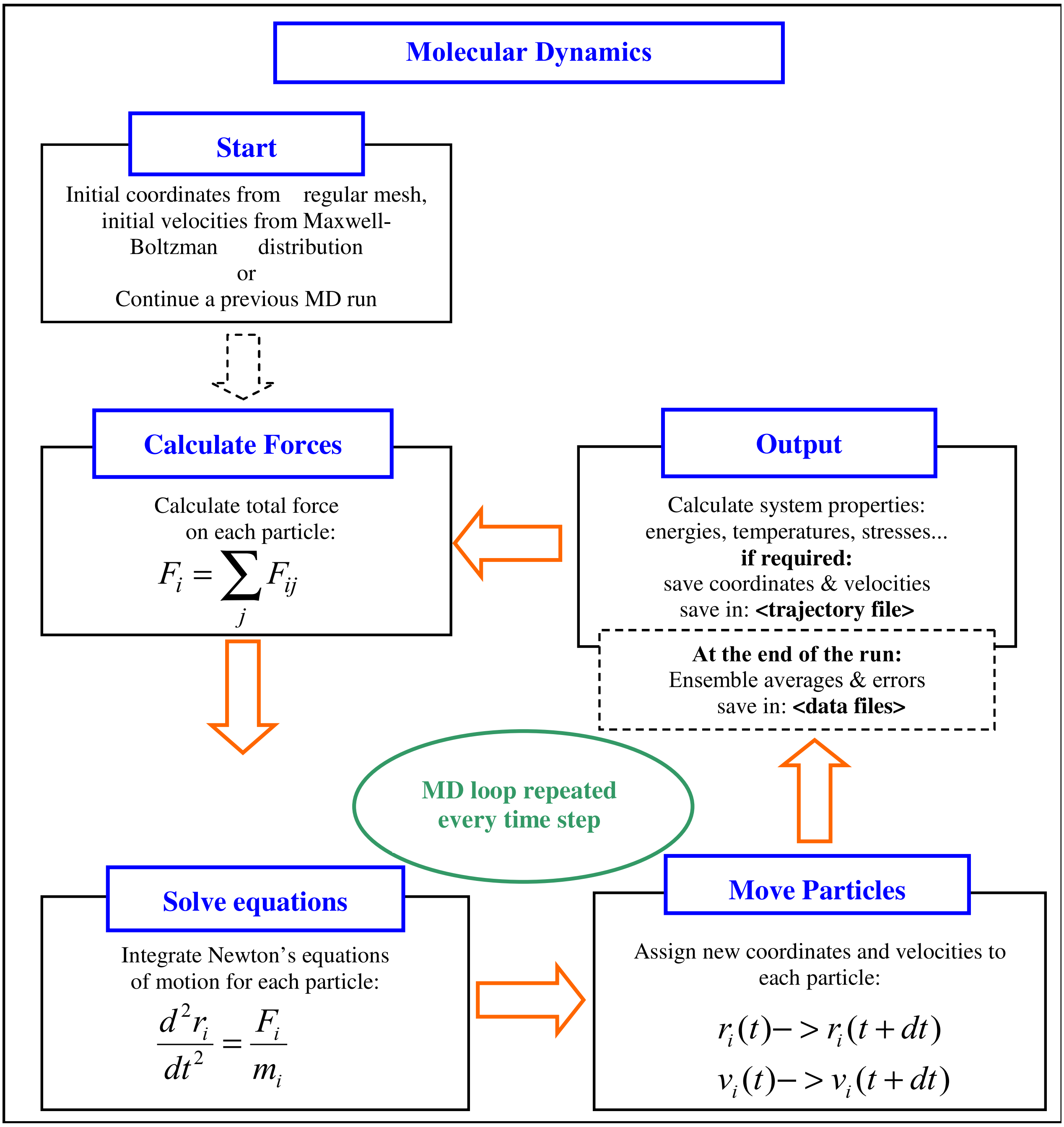

Molecular Dynamics
Start
Initial coordinates from regular mesh, initial velocities from Maxwell-Boltzman distribution
or
Continue a previous MD run
Calculate Forces
Calculate total force on each particle:
$F_i = \sum_j F_{ij}$
Output
Calculate system properties: energies, temperatures, stresses...
**if required:**
save coordinates & velocities
save in: **<trajectory file>**
**At the end of the run:**
Ensemble averages & errors
save in: **<data files>**
MD loop repeated every time step
Solve equations
Integrate Newton's equations of motion for each particle:
$\frac{d^2 r_i}{dt^2} = \frac{F_i}{m_i}$
Move Particles
Assign new coordinates and velocities to each particle:
$r_i(t) -> r_i(t+dt)$
$v_i(t) -> v_i(t+dt)$

Fig. 2 Simplified algorithm of Molecular Dynamics

$$M_i \ddot{R}_i = F_i + F_x$$

where $M_i$ is total mass molecule *i*, $R_i$ is the centre of mass of molecule *i*, $F_i$ is the total force acting on molecule *I*, $F_x$ is a mass force necessary to set water in motion (see Section IV).

## IV. Molecular dynamics simulations of plane nanoflows

In the present chapter the results of the simulations of plane flows of water through narrow copper channels [9,10,11,12] are presented.

Simulations were carried out for four models of water, as described in Section 2. The assumed channel width was equal to approximately 5 diameters of the water molecule. Physical properties of the materials and, in particular, their electrostatic interactions were taken into account.

To set water in motion, a mass force $F_x$, directed along x – axis, in positive direction, was applied to every water molecule. Three values of $F_x$ were selected: 0.5, 2.5 and 5.0. Mass force is nondimensional. This force set the only molecules of water. Wall of channel is stable during flows of water.

The simulations were performed with the program MOLDY [4], suitably modified for our purposes. Moldy is a computer program for performing molecular dynamics simulations of condensed matter. Moldy is free software; which may redistribute it and/or modify it under the terms of the GNU.

As mentioned before, some methods of temperature control may sometimes be necessary during the simulation. At the initial stage of equilibration the excess heat comes from potential energy of nonequilibrium configuration of the molecules (first 10000 time step - Fig 3 and Fig 4). Later, when

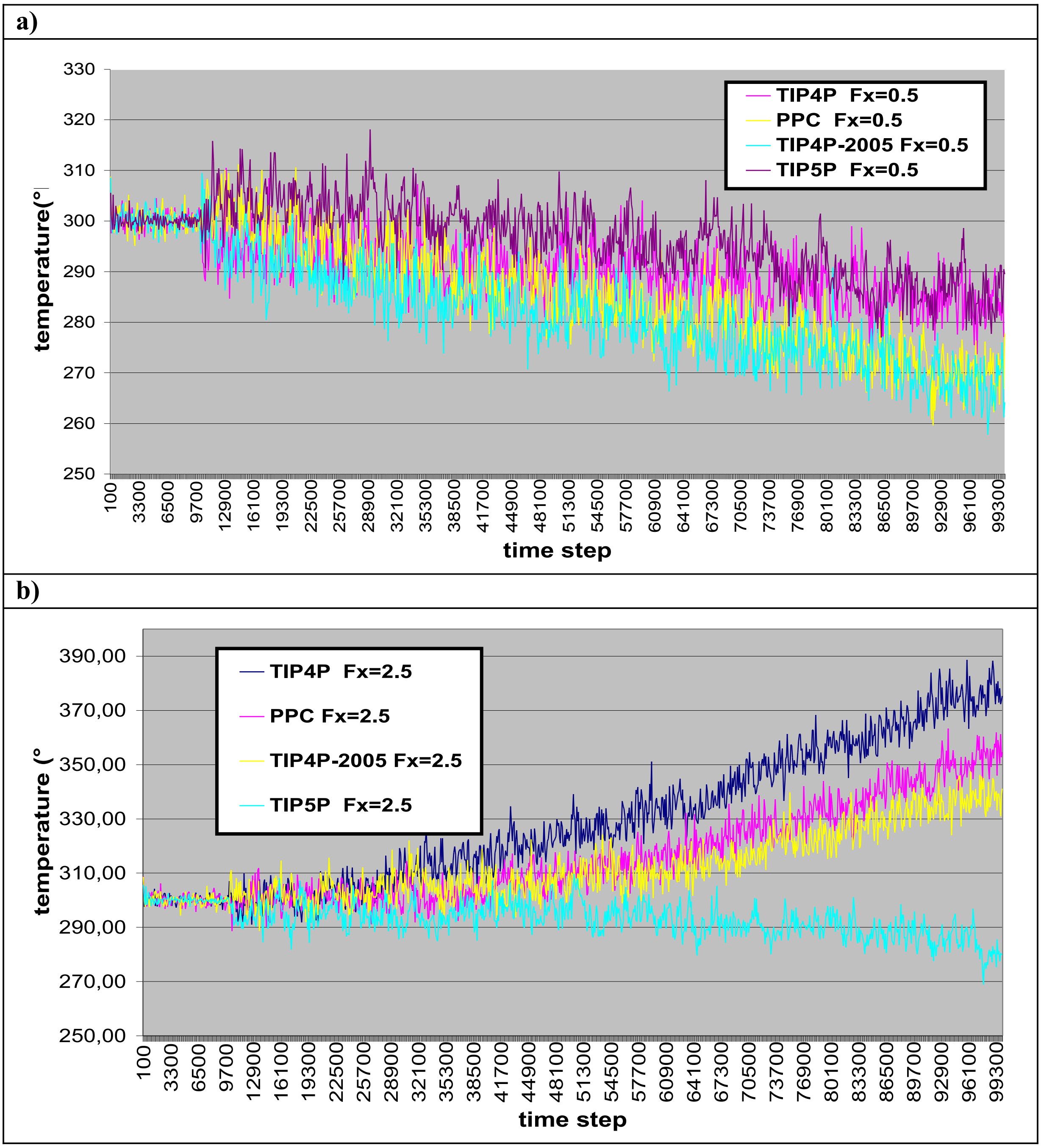


Fig. 3 Temperature of the system – Gaussian thermostat used for various models of water
a) mass force Fx = 0.5, b) mass force F x= 2.5.

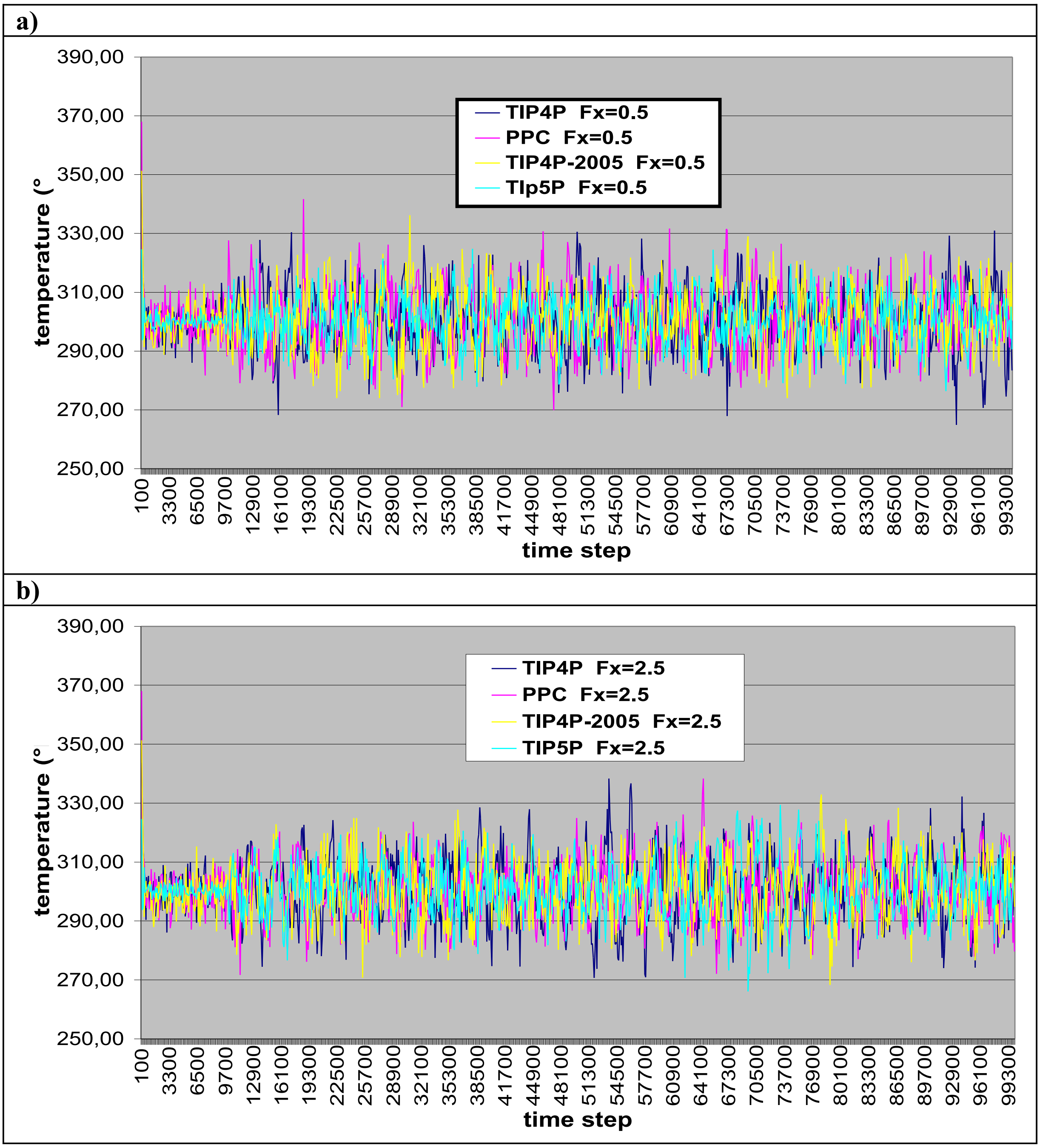


Fig. 4 Temperature of the system – Nosé-Hoover thermostat used for various models of water
a) mass force Fx = 0.5, b) mass force Fx = 2.5.

water flows under the influence of a mass force, heat is produced due to "friction" at the walls, which in molecular scale are always rough.

The program MOLDY offers three mechanisms of temperature control. In our simulations the technique of "velocity scaling" (multiplication of the molecular velocity components of all molecules by square root of the required temperature divided by the actual temperature of the medium) has been used during the equilibration period only, as it does not generate correct particle trajectories.

The Nosé-Hoover method couples the system to a heat bath using a fictional dynamical variable, while the Gaussian thermostat replaces the Newton-Euler equations by variants of which the kinetic energy is a conserved quantity [4]. The last two methods have been used to control temperature after equilibration of the system.

The calculations were carried out over 100 000 time steps **Δt**=0.005 picosecond long, after the system has reached the equilibrium. The positions and velocities of all molecules were recorded in dump files every 100 time steps, for further use.

Figures 3 – 7 present some selected results of our simulations. Figures 3 and 4 illustrate the efficiency of the two thermostats – Gaussian and Nosé-Hoover – for considered flows,

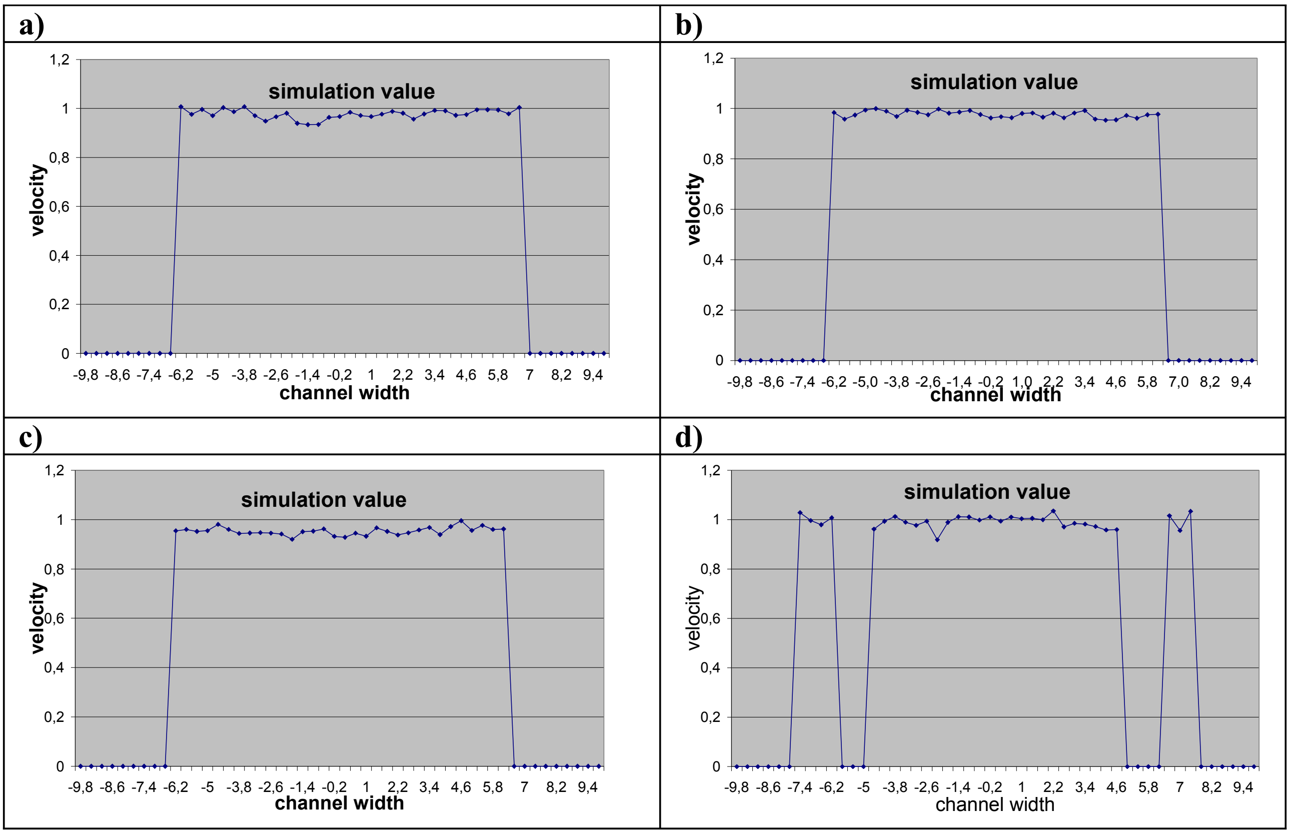


Fig. 5 Velocity distributions for flow of water in nanochannels, (Fx=0.5, Gaussian thermostat)
a) PPC, b) TIP4P, c)TIP4P-2005, d) TIP5P.

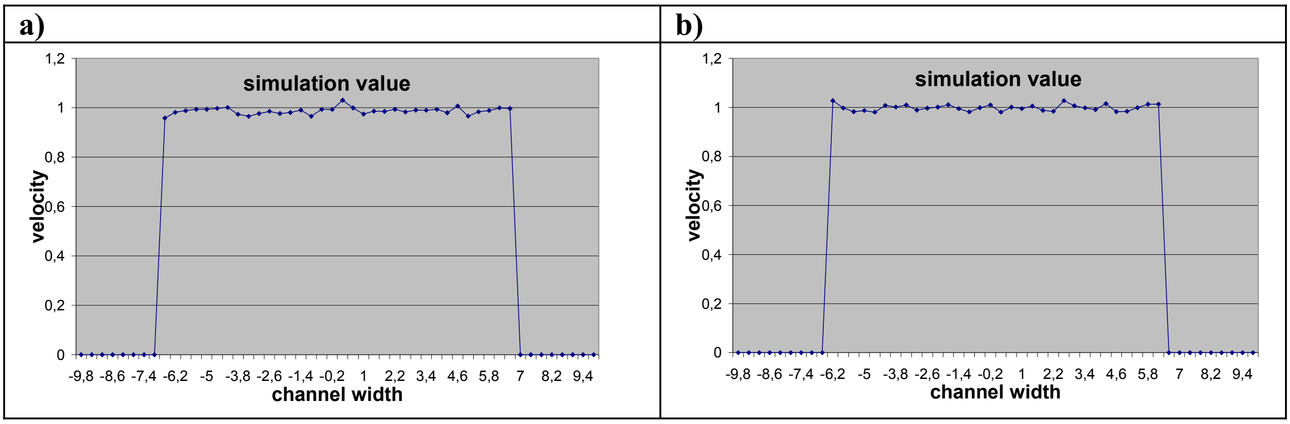


Fig. 6 Velocity distributions for flow of water in nanochannels, (Fx=0.5, Nosé-Hoover thermostat)
a) PPC, b) TIP4P.

at different mass forces acting on the molecules and different water models. On figure 3 we can see that for some the models of water the growth of force calling out the flow of water the growth of temperature effects.

Figures 5, 6 and 7 present velocity distributions in the channel cross-section perpendicular to the x – axis. Quickly increasing and declining graph it shows the wall of channel.

From the presented diagrams it is clear, that for the problems considered, i.e. flows in nanochannels, the Nosé-Hoover thermostat is much more efficient than the Gaussian one. The physical explanation of other peculiarities of the diagrams, particularly the behaviour of different models of water, requires further investigation. All our simulations be realized on our server on which programme MOLDY was compiled and started with different entrance described parameters overhead.

## References

[1] J. Grotendorst, D. Marx, A. Muramatsu: Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms, John von Neumann *Institute for Computing, Jülich, NIC Series,* Vol. 10, ISBN 3-00-009057-6, pp. 211-254, 2002.

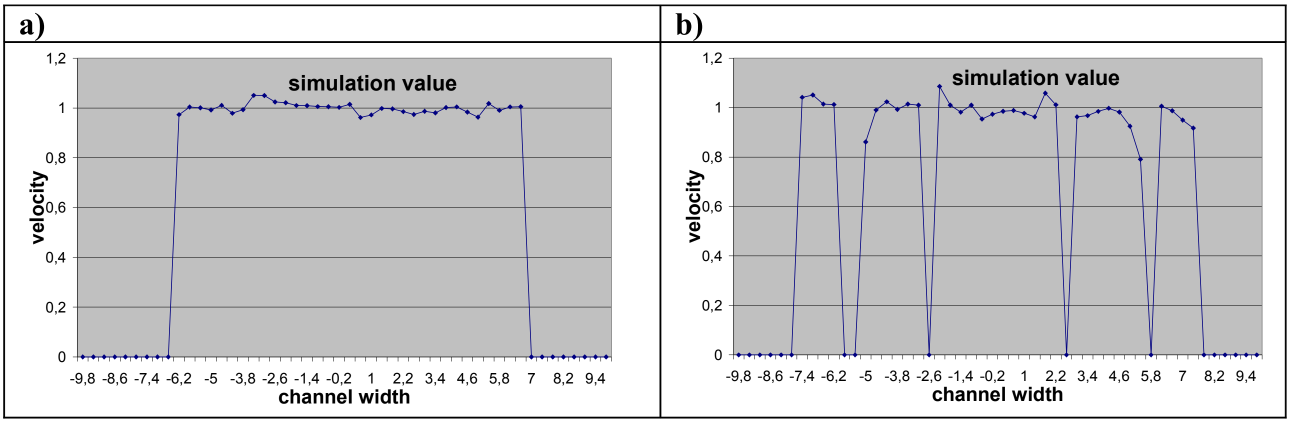


Fig. 7 Velocity distributions for flow of water in nanochannels, (Fx=0.5, Nosé-Hoover thermostat)
a)TIP4P-2005, b) TIP5P

[2] A. Z. Szeri, V. Radel: Flow modeling of thin films from macroscale to nanoscale, NATO Advanced Study Institute: Fundamentals of and Bridging the Gap between Macro- and Micro/nanoscale Tribology, Keszthely, Hungary, 2000.

[3] F. Ercolessi: A molecular dynamics primer, Spring College in Computational Physics, ICTP, Trieste, Italy, 1997.

[4] K. Refson: "Moldy User's Manual". Chapter II, ftp://ftp.earth.ox.ac.uk/pub.

[5] M. P. Allen, D. J. Tildesley: Computer simulations of Liquids, Oxford University Press, 1987.

[6] J. Adler: https//reu. Magnet.fsu.edu/program/2003/paper/adler;

[7] B.W. Van Beest, G. Kramer and R.A. Van Santen; Physical Rev. Letters 64, 16, 955-1975, 1990.

[8] J. Delhommelle, D. J. Evans: Molecular Physics, 100, 17, 2857-65, 2002.

[9] A. Kucaba-Piętal, Z. Walenta, Z.Peradzynski: TASK Quarterly, 5, 2, 179-189, 2001.

[10] A. Kucaba-Piętal.: Modelowanie mikroprzepływów na gruncie teorii płynów mikro-polarnych, Oficyna Wydawnicza Politechniki Rzeszowskiej, 2004.

[11] Z. Walenta, A. Kucaba-Piętal, Z. Peradzyński: Water flows in copper and quartz nanochannels, Mechanics XXI Century, Springer Netherlands, 2005;

[12] A. Kucaba-Piętal, Z. Peradzyński, Z. Walenta: Wall and size effect on water flows in nanochannels, ICTAM Proceedings 2005;

[13] M. Chaplin: Water Structure and Science, Water Models, http://www.lsbu.ac.uk/water/, 17.07.2008.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Sparse Shared-Memory Multifrontal Solver in SCAD Software

Sergiy Fialko
Cracow University of Technology, Poland ul. Warszawska 24, 31-155 Cracow, Poland
Email: sfialko@poczta.onet.pl

*Abstract*—**A block-based sparse direct finite element solver for commonly used multi-core and shared-memory multiprocessor computers has been developed. It is intended to solve linear equation sets with sparse symmetrical matrices which appear in problems of structural and solid mechanics. A step-by-step assembling procedure together with simultaneous elimination of fully assembled equations distinguishes this method from the well-known multifrontal solver, so this approach can be interpreted as a generalization of the conventional frontal solver for an arbitrary reordering. This solver is implemented in the commercial finite element software SCAD (www.scadsoft.com), and its usage by numerous users has confirmed its efficiency and reliability.**

## I. Introduction

THE solver is based on the idea of a step-by-step assembling of a given structure out of separate finite elements and substructures obtained at the previous assembling steps, and a simultaneous elimination of the fully assembled equations. It is an evolution of the frontal solver [1], and it differs from the classical multifrontal solver [2][3] which is a purely algebraic solver and gets the assembled global matrix in a compressed format as input data.

The key distinctive features of the presented method are:

- Each node is associated with a group of equations (usually, 6 equations per node for shell and space frame finite elements, and 3 equations per node for volumetric elements in unconstrained nodes). Thus, this approach produces the natural aggregation of equations, in this way improving the quality of reordering algorithms and speeding up the performance at the factorization stage. We refer to the elimination of a node of a finite element model at each elimination step. It means that a group of equations associated with the given node will be eliminated.
- An object-oriented approach is applied. A front is a C++ class object that encapsulates eliminated nodes at the current elimination step, a list of nodes and a list of equations for the current front, a pointer to the dense matrix which we denote as a frontal matrix, and so on.
- The whole elimination process is performed on a sequence of frontal matrices of the decreasing dimensionality.

Key points of the method are described in [4]. The paper [5] presents an improved version of the solver. The Cholesky block factorization algorithm was applied instead of a low-performance LU factorization from [4], and more efficient reordering algorithms comparing to [4] were implemented.

The current paper presents an extension of this solver onto the class of multi-core and multiprocessor shared-memory computers which are very popular today. The contemporary version of the solver is implemented in SCAD – one of the most popular finite element software applications for analysis and design of building structures in the Commonwealth of Independent States region.

## II. Analysis Stage

### A. Reordering and Symbolic Factorization

Modern reordering algorithms have a heuristic nature and do not provide an exact solution of the nonzero entries minimization problem. Moreover, we never know in advance what reordering strategy leads to the more optimal result for a given problem. Therefore several reordering methods have been developed. The most efficient methods for problems of structural mechanics usually are the minimum degree algorithm MMD , the nested dissection method ND , and the hybrid approach ND_MMD .

The fast symbolic factorization algorithm calculates the number of nonzero entries for each method of the ones listed above and then chooses the method that produces the least number of nonzero entries.

The use of a nodal adjacency graph instead of a graph of equations reduces the amount of data in a natural way and makes the reordering algorithms and the symbolic factorization procedure very fast: it takes only a few seconds even in very large problems (2,000,000 – 4,000,000 equations) to check 3 reordering methods and do the symbolic factorization.

### B. A Process Descriptor Data Structure

The reordering method establishes an order of the node elimination. The next step is to define the order of the finite element assembling.

The node is considered fully assembled if all finite elements, including this very node, have been already coupled. Adding any of the remaining finite elements does not make any change in the coefficients of equations associated with

This work was supported by the software company SCAD Soft (www.scadsoft.com)

this node. So, all equations belonging to the fully assembled node should be eliminated.

Let us consider a simple example, a square plate with the mesh 2x2 (Fig. 1). Before starting the elimination process, the whole structure is presented as exploded into separate finite elements.

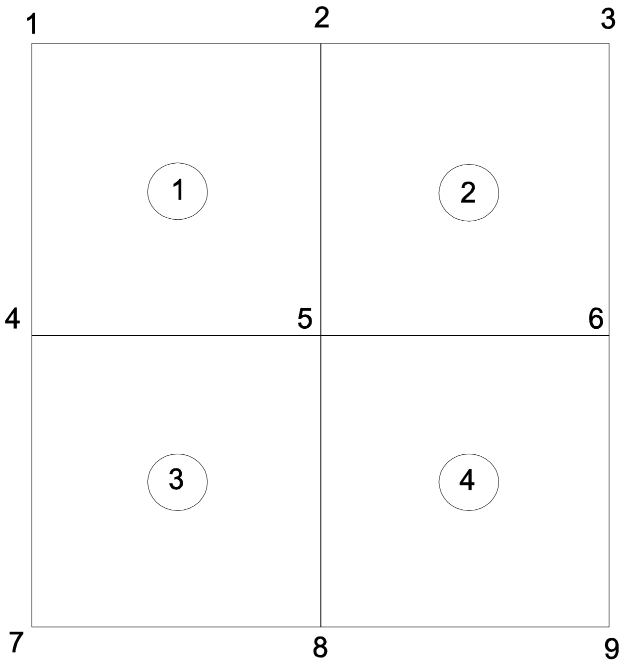


Fig. 1 Square plate 2x2

The sequence of nodal elimination produced by the reordering algorithm is: 1, 3, 7, 9, 2, 6, 8, 4, 5. Node 1 is the first node to be eliminated. This node belongs only to finite element 1. So, we take the finite element 1 and node 1 as fully assembled. The respective finite element matrix will be:

$$\begin{array}{cccc} 1 & 2 & 5 & 4 \\ \end{array}$$
$$\begin{pmatrix} s_{11} & & & \\ s_{21} & s_{22} & & \\ s_{31} & s_{23} & s_{33} & \\ s_{41} & s_{24} & s_{34} & s_{44} \end{pmatrix}, \qquad (1)$$

where $s_{\text{ij}}$ is a 6x6 matrix block because we suppose the design model to have six degrees of freedom per node. The global node numbers (global indexes) are shown at the top. The first block column contains the fully assembled equations. The partial factorization, covering only fully assembled equations, is then performed and the fully decomposed part of matrix is moved to a special buffer that contains the fully factorized global matrix. The remaining part of the matrix is an incomplete front; it will wait to be used at the following factorization steps. Local indexes are used during the partial factorization. Matrix (1) is a frontal matrix.

In the same way the nodes 3, 7, 9 are eliminated, and their respective incomplete fronts are created.

During the elimination of node 2 all finite elements have been already involved, and now we look through the lists of global indexes in the incomplete fronts. Node 2 is present in the global indexes of previous (incomplete) fronts 1, 2 (Table I), so we must assemble incomplete fronts 1, 2 to obtain a frontal matrix containing fully assembled equations for node 2.

This process is illustrated by Table I which shows a process descriptor data structure. The number of elimination steps is equal to the number of nodes of the design model.

We search for the node to be eliminated in the lists of nodes of each remaining finite element and in the lists of global indexes for each of the previous fronts. The fronts from the preceding elimination steps, which contain this eliminated node number, are the previous fronts. The previous fronts and their corresponding finite elements, pointed to by the last column of Table I, should be assembled to obtain the frontal matrix for the current front.

### C. Frontal Tree

The process descriptor data structure allows us to create a frontal tree. We take the last front from the bottom of Table I (it is Front 9) and put it at the top of the frontal tree (Fig. 2). Front 8 is the previous front for Front 9 – we put it under Front 9. For Front 8, Front 7 is its previous one – we put Front 7 under Front 8. And so on.

We reorder the fronts to reduce the storage memory required for incomplete fronts. The new front numbers are shown in italic under the original front numbers.

TABLE I.
A PROCESS DESCRIPTOR DATA STRUCTURE

| No. of front, elimination step | Node being eliminated | List of nodes in the frontal matrix | List of previous fronts | List of FEs fed to the assembling |
|---|---|---|---|---|
| 1 | 1 | 1,2,4,5 | — | 1 |
| 2 | 3 | 3,6,5,2 | — | 2 |
| 3 | 7 | 7,4,5,8 | — | 3 |
| 4 | 9 | 9,8,5,6 | — | 4 |
| 5 | 2 | 2,4,5,6 | 1,2 | — |
| 6 | 6 | 6,8,5,4 | 5,4 | — |
| 7 | 8 | 8,5,4 | 6,3 | — |
| 8 | 4 | 4,5 | 7 | — |
| 9 | 5 | 5 | 8 | — |

The frontal tree consists of (a) nodal fronts which have more than one previous front, (b) sequential fronts which have only one previous front, and (c) start fronts which do not have any previous front.

The core memory is allocated dynamically for objects of the nodal and start fronts. Each sequential front inherits the address of the frontal buffer from its previous front – this helps avoid the time-consuming memory allocation and copying of incomplete front. Moreover, if the sequence of sequential fronts does not have any assembled finite element, it is possible to consolidate such fronts to enlarge the block size of fully assembled equations and improve the performance. The consolidated frontal tree is presented in Fig. 3.

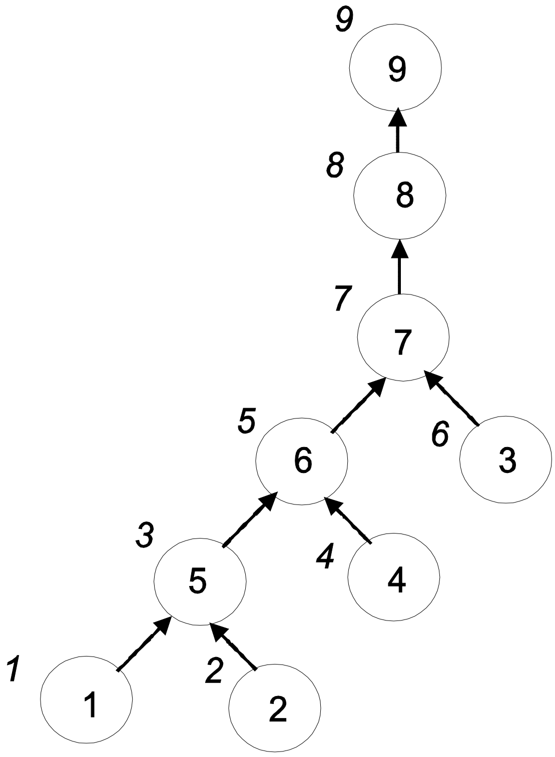


Fig. 2 A frontal tree

The buffer for storing the incomplete fronts (BuffFront), allocated in the core memory, is virtualized. When the size of the model exceeds the capacity of the random access memory (RAM), the buffer for the incomplete fronts is uploaded to hard disk. If all required previous fronts are in RAM during the assembling of the current nodal front, they are taken from BuffFront.

Otherwise, the previous fronts are taken from hard disk. The BuffFront is compressed from time to time to get rid of slow I/O operations during virtualization.

The presented method is a substructure-by-substructure approach, because each front presents a collection of coupled finite elements – a substructure, and the elimination process consists of a step-by-step assembling of these substructures and a simultaneous elimination of fully assembled equations.

## III. Factorization of Frontal Matrix

Generally, the structure of the frontal matrix can be very complex (Fig. 4). The matrix is symmetrical, and only the lower triangle part of it is stored in RAM. The fully assembled equations (grayed) are stored continuously but in arbitrary parts of the matrix (not necessarily at the top). This peculiarity restricts the direct application of the popular Lapack high-performance packages (Linpack, BLAS) and forces us to develop our own high-performance software for factorizations in frontal matrices.

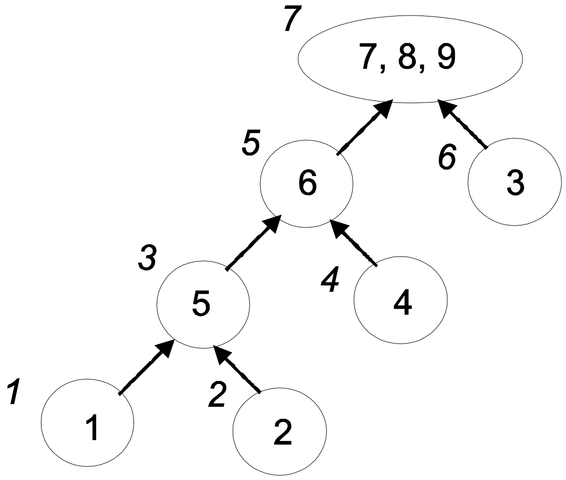


Fig. 3 Consolidated frontal tree

Thus, a Cholesky block method has been developed, which is a generalization of the frontal elimination approach [1], [4], where the fully assembled equations are stored in arbitrary parts of the matrix, onto the block version. It allows us to achieve the BLAS level 3 performance.

$$\begin{pmatrix} \mathbf{A} & \mathbf{W}_1^T & \mathbf{B}^T \\ \mathbf{W}_1 & \mathbf{F} & \mathbf{W}_2^T \\ \mathbf{B} & \mathbf{W}_2 & \mathbf{C} \end{pmatrix} =$$

$$= \begin{pmatrix} \tilde{A} & \tilde{W}_1 & \tilde{B}^T \\ 0 & L & 0 \\ \tilde{B} & \tilde{W}_2 & \tilde{C} \end{pmatrix} \cdot \begin{pmatrix} I & & \\ & S_L & \\ & & I \end{pmatrix} \cdot \begin{pmatrix} I & 0 & 0 \\ \tilde{W}_1^T & L^T & \tilde{W}_2^T \\ 0 & 0 & I \end{pmatrix} \qquad (2)$$

where $\mathbf{W}_1$, $\mathbf{F}$, $\mathbf{W}_2$ are blocks of fully assembled equations, **A, B, C** are blocks of partially assembled equations, which make up an incomplete front (Fig. 5) after the partial Cholesky factorization,

$$\begin{pmatrix} \tilde{\mathbf{A}} & 0 \\ \tilde{\mathbf{B}} & \tilde{\mathbf{C}} \end{pmatrix} \qquad (3)$$

The sign diagonal **I**, $\mathbf{S}_L$, **I** allows one to generalize the classic Cholesky factorization method onto a class of problems with indefinite matrices. The symbol "~" means that the corresponding matrix block is modified during the partial factorization.

The fully assembled blocks are moved to the buffer for the decomposed part of the matrix, and next the factorization of it is performed in the address space of this buffer.

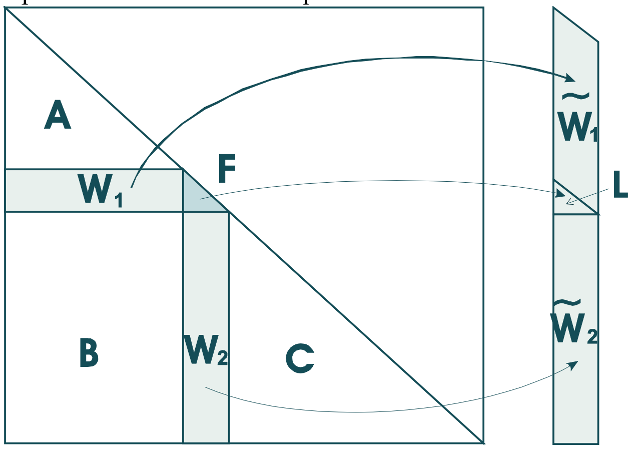


Fig. 4 The general structure of a frontal matrix. The fully assembled equations are moved to a buffer for the decomposed matrix

The incomplete front consists of sectors **A, B, C**. The symmetrical storage scheme is used. The coefficients of sector **A** remain in their positions, sector **B** moves up, and sector **C** moves up and to the left by the width of the grey strip. The final structure of the incomplete front is presented in Fig. 5.

The partial block factorization is performed in several steps:

- Factorize block **F**:

$$\mathbf{F} = \mathbf{L} \cdot \mathbf{S}_\mathrm{L} \cdot \mathbf{L}^\mathrm{T} \tag{4}$$

- Update blocks $\mathbf{W}_1$ , $\mathbf{W}_2$:

$$\begin{aligned} \mathbf{L} \cdot \mathbf{S}_L \cdot \widetilde{\mathbf{W}}_1^T &= \mathbf{W}_1 \rightarrow \widetilde{\mathbf{W}}_1^T \\ \mathbf{L} \cdot \mathbf{S}_L \cdot \widetilde{\mathbf{W}}_2^T &= \mathbf{W}_2^T \rightarrow \widetilde{\mathbf{W}}_2^T \end{aligned} \tag{5}$$

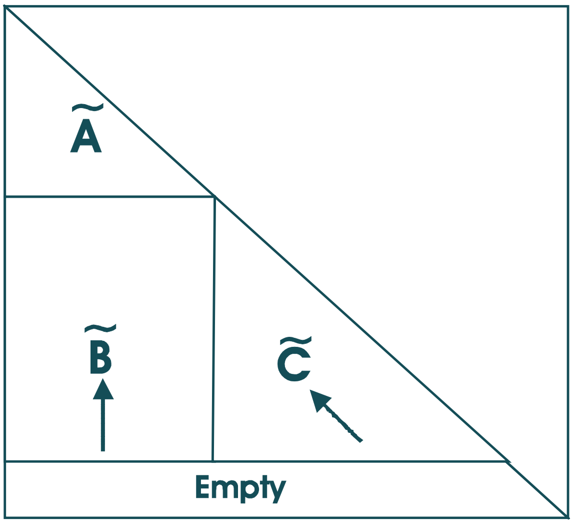


Fig. 5 The structure of the frontal matrix after the partial factorization. An empty strip appears at the bottom of matrix.

- Update sectors **A, B, C**:

$$\begin{aligned} \widetilde{\mathbf{A}} &= \mathbf{A} - \widetilde{\mathbf{W}}_1 \cdot \mathbf{S}_L \cdot \widetilde{\mathbf{W}}_1^T \\ \widetilde{\mathbf{B}} &= \mathbf{B} - \widetilde{\mathbf{W}}_2 \cdot \mathbf{S}_L \cdot \widetilde{\mathbf{W}}_1^T \\ \widetilde{\mathbf{C}} &= \mathbf{C} - \widetilde{\mathbf{W}}_2 \cdot \mathbf{S}_L \cdot \widetilde{\mathbf{W}}_2^T \end{aligned} \tag{6}$$

Expressions (4) – (6) are derived from (2) by means of the block matrix multiplication. Figs. 4, 5 help us to understand the structure of the frontal matrix. The main idea of the generalized Cholesky block factorization method is similar to the Gauss block elimination approach, reorganized to use the level 3 BLAS [10]. A particular case where a fully assembled part is at the top of the frontal matrix is presented in [5].

Matrix **F** has a small dimensionality, so there is no problem to store all matrix multipliers (4) in the processor cache and achieve a peak performance at this stage.

Stage (5) is a forward reduction performed on the package of right-hand sides – it is possible to achieve a good performance here, too.

The procedure that consumes most of the time is Stage (6). Each matrix multiplier $\mathbf{W}_1$, $\mathbf{W}_2$ is divided into smaller blocks, and the block matrix multiplication ensures the level 3 BLAS .

## IV. Parallel Implementation in Frontal Matrix

The bottleneck in the multi-core and multiprocessor shared-memory computing systems is an insufficient bandwidth of the system bus which leads to a weak speedup as the number of processors grows. A typical dependence for the speedup parameter, $S_p = T_1 / T_p$ , where $T_1$ is a computing time on one processor, $T_p$ is that on $p$ processors, for algorithms where the performance factor $q \le 2$ (matrix-vector multiplication, non-block matrix-matrix multiplication) is presented in Fig. 6. The performance factor is $q = f/m$ where $f$ is the number of arithmetic operations for a given algorithm and $m$ is the number of data transfers between RAM and the processor cache.

A totally different dependence takes place on the same computer for the block matrix-by-matrix multiplication; here $q \sim \sqrt{M}$ where $M$ is the cache size (fig. 7).

So, it is possible to achieve an efficient speedup for computers with an insufficient bandwidth of the system bus by increasing the number of processors for level 3 BLAS algorithms, where $q \sim \sqrt{M}$ , because these algorithms use significantly fewer RAM – cache – RAM transactions per single arithmetic operation than the algorithms where $q \le 2$ .

The results presented in Figs. 6 through 8 have been obtained by special tests prepared by the author.

Therefore the main attention during the development of the parallel code is paid to the Cholesky block factorization in the frontal matrix and the fork-joint parallelization technique, which is used mainly at the stage 6 during the block matrix-by-matrix multiplication.

The Microsoft Visual Studio 2008 environment (the C++ language for most of the code and the C language for “bottleneck” fragments of the code in order to achieve a peak efficiency by compiler optimizations) with the OpenMP support is used. All program code, including the factorization of frontal matrix, was developed by the author.

The local reordering of equations performed in the frontal matrix for the nodal and start fronts allows us to reduce the amount of time-consuming operations of moving sectors **B** , **C** block and thus increase the performance of method (Fig. 8).

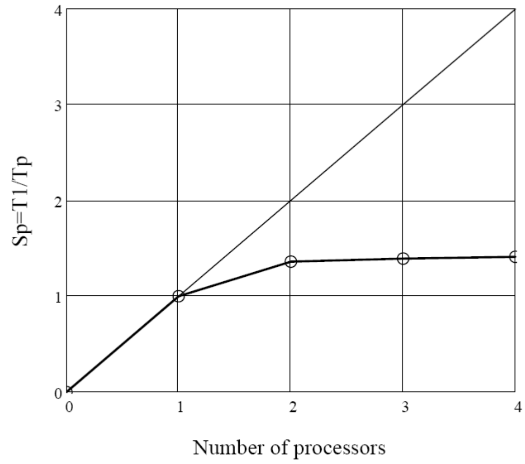


Fig. 6. $Sp = T_1 / T_p$ vs. number of processors $p$ for algorithms where the performance factor $q \leq 2$

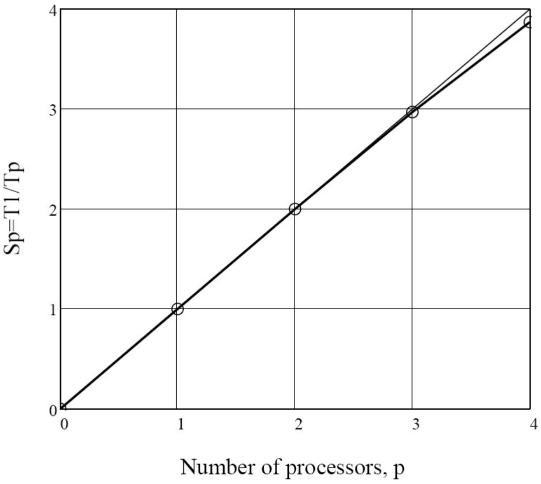


Fig. 7. $Sp = T_1 / T_p$ vs. number of processors $p$ for algorithms where the performance factor $q \sim M^{1/2}$

We reorder the frontal nodes according to the sequence of global reordering and then take the reverse order – each eliminated node occurs at the end of the node list. It ensures such a structure of the frontal matrix that the fully assembled equations are at the bottom and no moving of incomplete fronts is required (only sector A is shown in the figure, sectors B, C are omitted).

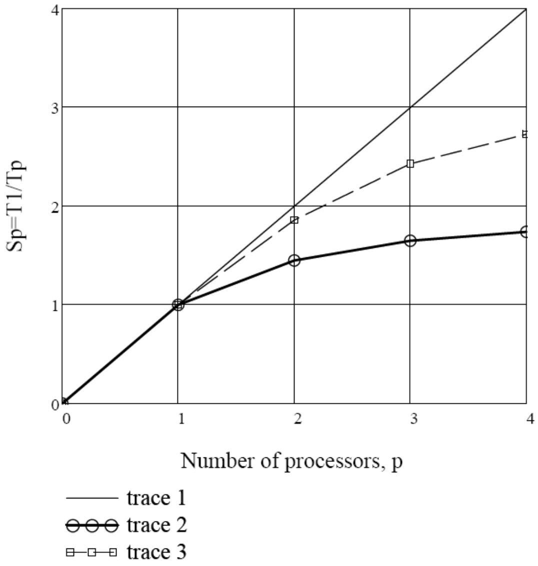


Fig. 8. $Sp = T_1 / T_p$ vs. number of processors $p$ for factorization of frontal matrices. Trace 1 – perfect speedup, trace 2 – local fronts not reordered, trace 3 – reordered fronts.

For instance, in the example of Fig. 1 the order of node processing for front 7 (tab. I) must be 5, 4, 8 rather than 8, 5, 4, because this numbering ensures that at each front elimination step the fully assembled equations stay at the bottom of the matrix and no moving of incomplete fronts is required.

Typical dependencies $S_p = T_1 / T_p$ vs. the number of processors $p$ for factorization of frontal matrices for presented method are shown in Fig. 8. The total time of factorization for all frontal matrices is under consideration. An essential improvement of performance occurs when the local reordering of equations in frontal matrices is performed (the dashed curve corresponds to local reordering and solid curve to no local reordering).

## V. Numerical Results

All results presented further below have been obtained on the computer Intel® Core™2 Quad CPU Q6600 @2.40 GHz, cache 4xL1: 32 KB (code), 4xL1: 32 KB (data), 2xL2:4096 KB, random-access memory DDR2 800 MHz 4 GB, operating system Windows XP Professional.

### *D. Test 1*

A cube with the mesh 40x40x40 consisting of brick volumetric finite elements is considered. Four corner nodes of the bottom face are restrained. This model comprises 68 921 nodes, 64 000 finite elements, 206 751 equations; the whole model is stored in RAM. The analysis stage, which includes reordering, symbolic factorization, creation of the process descriptor data structures, reordering and consolidation of the frontal tree, is performed as a serial computation. The computing time is 3 s. The solution stage, including forward – backward reductions, takes 4 s. The parallel computing stage covers only the factorization of frontal matrices and takes a predominant part of the whole time required by the numerical factorization stage (Table II).

The factorized matrix, as well as the buffer for incomplete fronts, is stored in RAM, and due to this fact the speedup parameter on four processors is about 3.0. It is a good result for multi-core and shared-memory office computers. For problems of structural mechanics solved by finite element software based on the Intel Math Kernel Library (PARDISO solver –[15]), the speedup parameter on four processors is about 2.9.

Table II.
Time of Numerical Factorization for Cube 40x40x40

| Number of processors | Numerical factorization – $T_p$, s | $S_p = T_1/T_p$ |
|---|---|---|
| 1 | 519 | 1.0 |
| 2 | 280 | 1.85 |
| 3 | 202 | 2.57 |
| 4 | 167 | 3.10 |

However, the PARDISO solver works only with the core memory, so its capabilities are restricted to problems of relatively small dimensions.

### *B. Test 2*

A square plate with two meshes 400x400 and 800x800 is considered. A quadrilateral flat shell finite element is used. The time of numerical factorization was compared with that for the sparse direct solver from ANSYS v 11.0. The first problem (mesh 400x400, Table III) contains 964 794 equations and the second one (mesh 800x800, Table IV) contains 3 849 594.

Table III.
Time of Numerical Factorization for Plate 400x400

| Number of processors | Numerical factorization for ANSYS v. 11.0, s | Numerical factorization for presented solver, s |
|---|---|---|
| 1 | 221 | 226 |
| 2 | 176 | 152 |
| 4 | 159 | 121 |

Table IV.
Time of Numerical Factorization for Plate 800x800

| Number of processors | Numerical factorization with ANSYS v. 11.0, s | Numerical factorization with the presented solver, s |
|---|---|---|
| 1 | failed | 2 091 |
| 2 | failed | 1 450 |
| 4 | failed | 1 080 |

For the first model (mesh 400x400), our solver stores the factorized matrix on hard disk, but the buffer for incomplete fronts is stored in the core memory. Due to virtualization of the factorized matrix, the scalability is worse comparing to the previous model – the speedup parameter on four processors is about 1.9. The performance demonstrated by both the sparse direct solver from ANSYS v. 11.0 and our solver is about the same.

For the second model (mesh 800x800), the solver from ANSYS v.11.0 failed due to insufficient RAM. Results for our solver are presented in Table IV. The factorized matrix is stored on hard disk, and virtualization is used for the buffer that contains incomplete fronts. The scalability is weak — the speedup parameter on four processors is about 1.9. The size of the factorized matrix is 9 723 MB.

### *C. Test 3*

The design model of a multistory building complex is presented in Fig. 9.

The number of equations is 1 534 674, the size of the factorized global matrix is 5 355 MB, the numerical factorization time on four processors is 860 sec. The disk memory is used for virtualization of the factorized matrix as well as for virtualization of the buffer where the incomplete fronts are stored.

Test A demonstrates that the scalability of the presented method for multi-core and multiprocessor office computers is similar to the scalability of the sparse direct solver based on Intel Math Kernel Library when the dimensionality of the model permits to use only the core memory. Test B proves that the performance of our solver is similar to the performance of the conventional multifrontal solver from the well-known ANSYS software. In addition, test B shows that the presented solver works very efficiently with the core memory. Test C illustrates that the presented method solves a really large finite element problem of structural mechanics very efficiently.

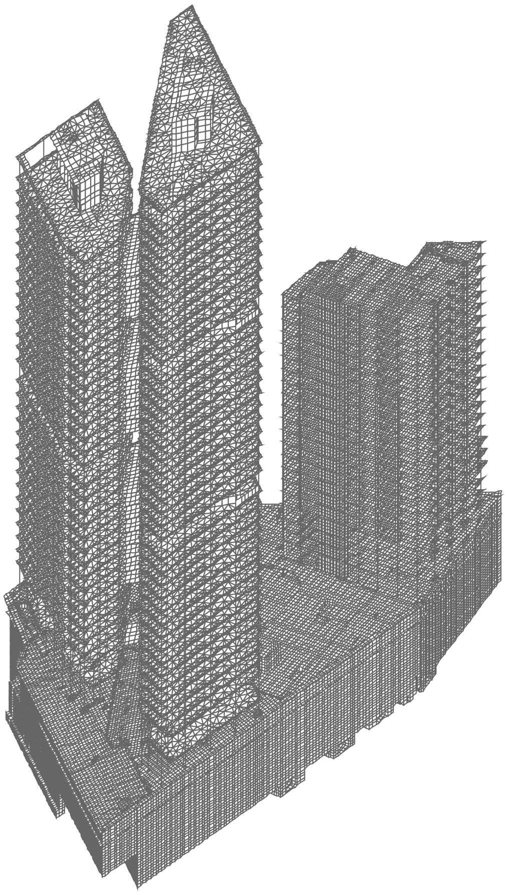


Fig. 9. Finite element model of a multistory complex “Moscow-city”, 255 779 nodes, 347 767 finite elements, 1 534 674 equations

## VI. Conclusion

A block-based sparse direct multifrontal solver for finite element problems of structural and solid mechanics has been developed. This method is oriented mainly at the usage on common shared-memory multiprocessor and multi-core computers.

The proposed solver has a clear mechanical interpretation as a substructure-by-substructure method. It uses the modern reordering algorithms which effectively reduce the fill-ins.

The Cholesky block algorithm with a generalization onto the class of indefinite matrices is applied here to the factorization of frontal matrix. The complicated structure of the frontal matrix restricts the direct application of the BLAS high-performance package and forces us to develop our own high-performance code based on a subdivision of the frontal matrix into blocks.

The OpenMP technique is applied to produce a fork-joint parallelization. The block subdivision and an appropriate numbering of equations in the frontal matrix, which reduces the amount of low-performance moving operations, allow us to improve the scalability of the method.

A comparison with the sparse direct solver from ANSYS v11.0 demonstrates that the performance of the proposed solver is close to the performance of ANSYS, but the proposed solver allows us to solve essentially larger problems on the same computers than the ANSYS solver is capable of processing.

The reliability and efficiency of the proposed solver has been confirmed by numerous users of the SCAD commercial software.

## References

[1] B. M. Irons, “A frontal solution program for finite-element analysis”, *International Journal for Numerical Methods in Engineering*, 2, pp. 5 –32, 1970.

[2] I. S. Duff, J. K. Reid, “The multifrontal solution of indefinite sparse symmetric linear systems”, *ACM Transactions on Mathematical Software*, 9, pp. 302–325, 1983

[3] F. Dobrian, A. Pothen, “Oblio: a sparse direct solver library for serial and parallel computations”, *Technical Report describing the OBLIO software library*, 2000.

[4] S. Yu. Fialko, “Stress-Strain Analysis of Thin-Walled Shells with Massive Ribs”, *Int. App. Mech.*, 40, N4, pp. 432–439, 2004.

[5] S. Yu. Fialko, “A block sparse direct multifrontal solver in SCAD software”, in *Proc. of the CMM-2005 – Computer Methods in Mechanics June 21-24, 2005*, Czestochowa, Poland, pp. 73 – 74.

[6] A. George, J. W.-H. Liu, “The Evolution of the Minimum Degree Ordering Algorithm”, *SIAM Rev.*, 31, March, pp. 1-19, 1989.

[7] A. George, J. W.-H. Liu, *Computer solution of sparse positive definite systems* , New Jersey : Prentice-Hall, Inc. Englewood Cliffs, 1981, ch. 8.

[8] C. Ashcraft, J. W.-H. Liu, “Robust Ordering of Sparse Matrices Using Multisection”, *Technical Report CS 96-01, Department of Computer Science, York University, Ontario, Canada,* 1996.

[9] A. George, J. W.-H. Liu, *Computer solution of sparse positive definite systems*, New Jersey : Prentice-Hall, Inc. Englewood Cliffs, 1981, ch. 5.

[10] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, Philadellphia, 1997, ch. 2.

[11] A. Kumbhar, K. Chakravarthy, R. Keshavamurthy, G. V. Rao, “Utilization of Parallel Solver Libraries to solve Structural and Fluid problems”, White paper by Cranes Software, http://www.intel.com/cd/software/products/asmo-na/eng/373466.htm.

[12] N. I. M. Gould, Y. Hu, J. A. Scott, “A numerical evaluation of sparse direct solvers for the solution of large sparse, symmetric linear systems of equations”, *Technical report RAL-TR-2005-005, Rutherford Appleton Laboratory, 2005.*

[13] O. Schenk, K. Gartner, “On fast factorization pivoting methods for sparse symmetric indefinite systems”, Technical Report CS-2004-004, Department of Computer Science, University of Basel, Switzerland, 2004.

[14] O. Schenk, K. Gartner, “Solving unsymmetric sparse systems of linear equations with PARDISO”, Journal of Future Generation Computer Systems, 20(3), 475–487, 2004.

[15] O. Schenk, K. Gartner, W. Fichtner. “Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors”, BIT, 40(1), 158–176, 2000.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 285–291

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Testing Tesla Architecture for Scientific Computing: the Performance of Matrix-Vector Product.

Paweł Macioł
Cracow University of Technology,
Insitute of Computer Modelling,
ul. Warszawska 24, 31-155 Kraków, Poland
Email: pmaciol@pk.edu.pl

Krzysztof Banaś
AGH University of Science and Technology,
Department of Applied Computer Science and Modelling,
al. Mickiewicza 30, 30-059 Kraków, Poland
Cracow University of Technology,
Insitute of Computer Modelling,
ul. Warszawska 24, 31-155 Kraków, Poland
Email: pobanas@cyf-kr.edu.pl

***Abstract*—The paper presents results of several experiments evaluating the performance of NVIDIA processors, implementing a new Tesla architecture, in matrix-vector multiplication. Three matrix forms, dense, banded and sparse, are considered together with three hardware platforms: NVIDIA Tesla C870 computing board, NVIDIA GeForce 8800 GTX graphics card and one of the newest Intel Xeon processors, E5462, with 1.6 GHz front side bus speed. The conclusions from experiments indicate what speed-ups can be expected when, instead of standard CPUs, accelerators in the form of presented GPUs are used for considered computational kernels.**

## I. Motivation

THE USE of graphics processing units (GPUs) in scientific computing is becoming an accepted alternative for calculations employing traditional CPUs [1]. The characteristics of GPUs, especially parallel execution capabilities and fast memory access, render them attractive in many application areas. One of the most important application domains is numerical linear algebra. The computational kernels from linear algebra are used in many scientific codes. Hence the widespread interest in porting and testing such kernels for GPUs [2].

The purpose of the present article is to assess the performance of the recent NVIDIA GPUs in performing one of linear algebra kernels, namely matrix-vector product. This kernel plays an important role in the implementation of iterative solvers for systems of linear equations. Moreover it is a typical memory-bound operation—its performance depends mainly on the speed of communication between a processor and memory chips, much less on the processing capabilities of the processor itself.

The organization of the paper is the following. In the next section performance characteristics of NVIDIA GPUs are described and compared to characteristics of typical contemporary CPUs. Section III presents the matrix formats considered in the paper and the corresponding matrix-vector multiplication algorithms. In Section IV the set-up of experiments as well as tests' results are described. Finally, conclusions are drawn in Section V.

## II. Performance characteristics of CPUs and GPUs

A typical contemporary processor is a two- or four-core unit equipped with a memory hierarchy comprised of several layers of cache and the main memory. From the point of view of performance for scientific codes two parameters are of premium importance: the processing speed and the speed of data transfer from the memory.

The processing speed depends on the number of cores, clock frequency and the number of instructions completed in every clock cycle. This last number varies greatly depending on the application. The theoretical maximum is usually two to four instructions per cycle. The practical performance can be closed to the maximum whenever "the memory wall" is not hit, i.e. the processor gets all necessary data on time. This is the case for BLAS Level 3 routines, like e.g. matrix-matrix product, on which the direct solution of systems of linear equations is usually based.

There is a different situation with memory bound algorithms. If the processor cannot get the necessary data on time the performance can be equal to a small percentage of the maximum.

Graphics processing units differs significantly from general purpose CPUs in both aspects affecting performance. Their processing speed is much greater due to the large number of specialised cores (though usually operating at lower frequencies that CPU cores). Also the throughput to the memory is greater for GPUs due to usually wider buses then that of CPUs. Hence both, processing speed limited and memory limited algorithms can benefit from off-loading to GPUs. One of serious drawbacks of contemporary GPUs is the use of single precision floating point numbers only. However, today all major producers of GPUs aiming at general purpose computing start offering double precision floating point capabilities in their products, so this limitation should shortly be overcome.

### A. An example CPU

Let us take, as an example CPU, one of the newest Intel Quad-core Xeon processors, E5462 with four cores, 2.8 GHz

**Algorithm 1** Simple matrix-vector product, $y := A \cdot x$, for dense matrices stored column-wise

```
for(i=0;i<n;i++){
   t:=x[i];
   for(j=0;j<n;j++){
      y[j]+=A[i*n+j]*t;
} }
```

clock speed and four double precision floating point instructions per clock cycle (maximum). The theoretical peak floating point performance is hence 44.8 GFlops. The processor is equipped with 12 MB cache memory and 1.6 GHz front side bus.

Let us consider a simple example of matrix-vector multiplication, a BLAS Level 2 operation that is memory bound. For a square matrix of size $n$ the number of floating point operations performed is $2 * n^2$. The number of memory accesses depends on the details of the algorithm. Let us consider a simple algorithm for column-wise stored matrix $A$ (Algorithm 1 on page 286). The minimal number of memory accesess (e.g. for small matrices fitting in cache together with vectors $x$ and $y$) is equal to $n^2 + 3 * n$ (one read for every element of $A$, $x$ and $y$ and one write for every element of $y$). The maximal number of memory accesses (e.g. for large matrices when at the end of the inner loop the first elements of $y$ are no longer in cache memory of any level) is $3n^2 + n$ (one read for every element of $A$, $n$ reads and $n$ writes for each element of $y$ and one read for every element of $x$). The number of memory accesses per one floating point operation varies, hence, between $\frac{1}{2} + \frac{3}{2*n}$ and $\frac{3}{2} + \frac{1}{2*n}$.

The theoretical peak throughput to the memory for the E5462 processor is 12.8 GB/s (1.6 GHz FSB speed combined with 8 bytes wide bus). This translates to 1.6 billions double precision numbers supplied to the processor per second. For the considered matrix-vector product algorithm this means (taking into account the number of memory accesses to the number of operations ratio and neglecting the terms with $n$ in the denominator) the constraint on the performance at the level between 3.2 Gflops and 1.067 Gflops. These constitutes only 7.14% and 2.38%, respectively, of the theoretical peak performance of the processor.

### B. An example GPU

Graphics processors have evolved in recent years from special purpose devices, with a sequence of fixed-function elements performing subsequent phases of the standard graphics pipeline, to processors that, although keeping many of special purpose solutions, can serve general purpose computations. NVIDIA, being the company that introduced the first GPU in 1999, has recently developed a special architecture that unifies the hardware for different stages of graphics processing and at the same time offers general purpose computing capabilities. The Tesla architecture is depicted in Fig. 1 (taken from [3], from which also the presented below facts about the Tesla architecture are taken).

From the general computing point of view the important elements of the architecture are the following:

- Host interface and Compute work distribution—the elements responsible for getting instructions and data from the host CPU and its main memory; they also manage the threads of execution by assigning groups of threads to processor clusters and performing contex switching
- TPC—texture/processor cluster, a basic building block of Tesla architecture GPUs (a single GPU can have from one to eight TPCs)
- SM—streaming multiprocessor, a part of TPC, consisting of eight streaming processor cores, two special function units (for performing interpolation and approximate evaluation of trigonometric functions, logarithms, etc.), a multithreaded instruction fetch and issue unit, two caches and a pool of shared memory
- Texture unit—each unit is shared by two SMs (each TPC contains two SMs and one texture unit), the unit plays its part in graphics calculations and is equipped with L1 cache memory accessible from SPs
- Level 2 Cache memory units—connected through fast network to TPCs

(ROP—raster operation processors usually do not take part in general purpose computations)

The most important characteristic of Tesla processing is that streaming multiprocessors manage the execution of programs using so called “warps”, groups of 32 threads. All threads in a warp execute the same instruction or remain idle (in such a way different threads can perform branching and other forms of independent work). Warps are scheduled by special units in SMs in such a way that, without any overhead, several warps execute concurrently by interleaving their instructions. So it is possible that, for example, with two warps the execution looks as follows: warp_1-instruction_1, warp_2-instruction_1, warp_1-instruction_2, warp_1-instruction_3, warp_2-instruction_2, etc. In such a way each SM can manage up to 24 warps, i.e. 764 threads. From the point of view of writing application codes it is, however, important to take into account the organization of the work, i.e. the use of 32 threads simultaneously. The code that does not break into 32 thread units can have much lower performance.

The processing capabilities of Tesla GPUs derive from many sources. Some of them, like texture and rasterization units, serve mainly the purposes of graphics processing. For general processing, especially scientific array operations, the main units are streaming processor cores (SP) and special function units (SFU). Each core can perform two floating point operations per cycle (by the multiply-add unit), each SFU is equipped with four multipliers, hence can perform 4 instructions per cycle. This gives 16 operations per cycle for 8 SPs and 8 operations per cycle for two SFUs in a single streaming multiprocessor. If an application can make both types of units work simultaneously (which is possible theoretically) it can achieve 24 operations per cycle per SM,

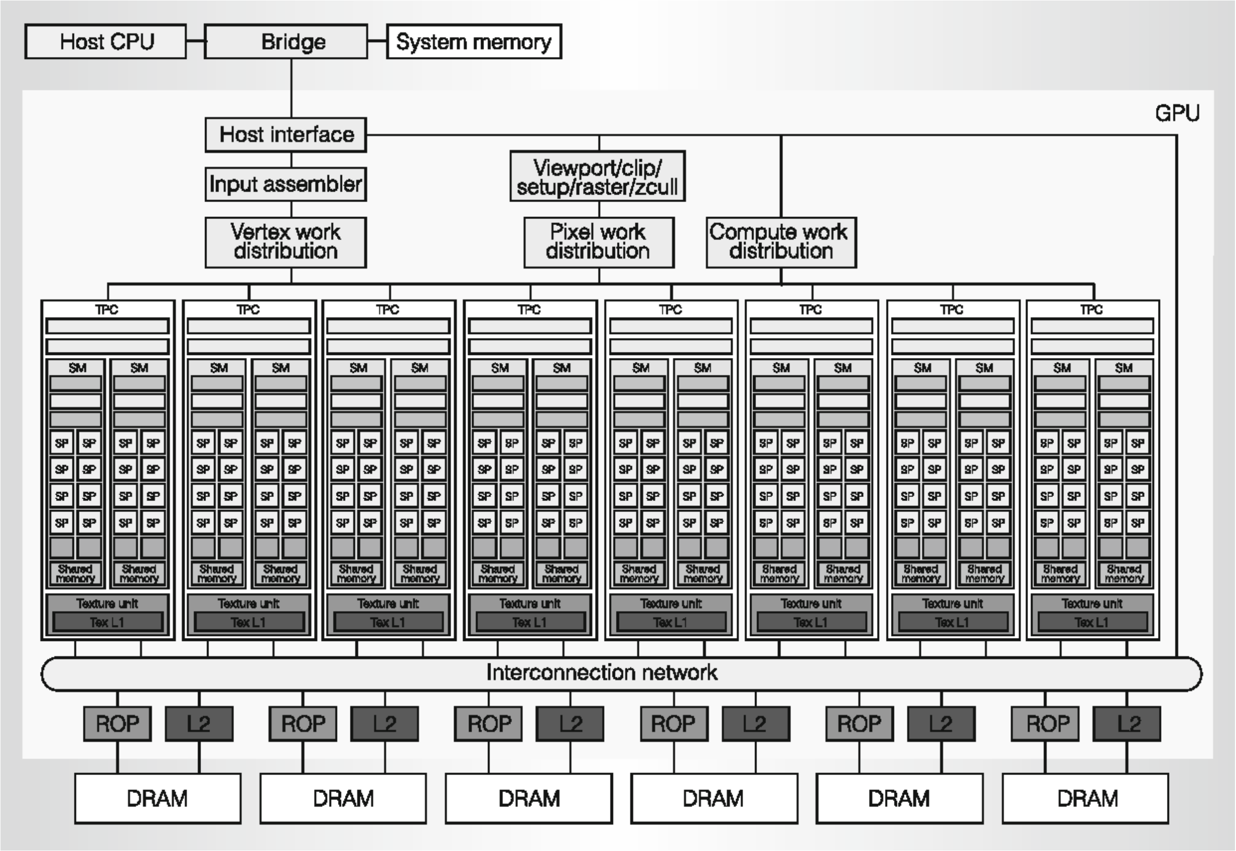


Fig. 1. Tesla architecture [3].

that for 1.5 GHz GPU with 16 SMs gives 576 Gflops. The performance possible to obtain in real applications is, however, much smaller.

One of reasons for smaller performance can be again limitations caused by too slow memory access. The memory system of Tesla architecture consists of even more layers than that of a CPU. Apart from the main memory of the host system, there is a DRAM card memory and two pools of cache memory - one within SMs, called shared memory, and the second within texture units. Both caches can be utilized by scientific codes. Important feature of the architecture is that the processor is not a vector unit, each thread issues its own memory requests. However, the performance is best when requests can be blocked by memory management unit to access contiguous memory areas.

The DRAM memory of a card consist of up to six modules, that comprise the whole physical memory space of a GPU. Each module is connected to the GPU by a link 64-bits wide. Combined with the DDR technology and clock frequency in the range of 1GHz this gives the memory throughput around 100 GB/s. This is four times faster than the theorethical memory throughput of the example Xeon CPU considered earlier. In our simple example of matrix-vector multiplication this translates to the speed between 37.5 and 50 Gflops, but now for single precision numbers only. Still this constitutes only less than 10% of the theorethical GPU performance.

### C. CUDA programming model

CUDA programming model is a model for programming Tesla GPUs using some extensions of standard C/C++. The extensions include two main domains: parallel work organisation through concurrent threads and memory accesses making use of the memory hierarchy of Tesla architecture.

Threads in the CUDA model are grouped into so called thread blocks. A programmer selects the block size. All threads in a block executes on one SM and can use its shared memory. Threads in one block can communicate with each other using the shared memory. Threads in different blocks cannot communicate. This constraint has also some positive consequenses since scheduling of different blocks is fast and flexible (independent of the number of SMs used for program execution).

Apart from shared memory variables, the programmer can explicitly address two other types of variables. Local variables reside in DRAM card's memory and are private for every thread. Global variables also reside in DRAM card's memory and are accessible to threads from different blocks, thus providing i.e. a way for global thread synchronisation. The DRAM memory is, however, much slower than the on-chip shared memory. For this reason threads within blocks can synchronise using a special instruction implemented using the shared memory.

Writing programs in CUDA consist in writing two types of routines. Standard routines are executed on CPU. From these routines kernel procedures are invoked, to be executed by the GPU. Kernels are written as procedures (declared with a keyword `__global__`) for one thread. Parallel execution is obtained by specifying in the invocation, using a special syntax, the number of thread blocks and the number of threads in each block. For our example of matrix-vector product the calling sequence in a program executed by CPU might look as follows:

```
dim3 threads( BLOCK_SIZE )
dim3 grid( n/BLOCK_SIZE );
mv_kernel<<< grid, threads >>>
  ( A_G, x_G, y_G, n );
```

where `threads` is a special variable of type `dim3` that has three components, each for one dimension of a thread block. Dimensions are used for identifying threads within blocks (in our example each thread is uniquely tagged by a triple `threads.x, threads.y, threads.z`, the first one within the range $(0, BLOCK\_SIZE - 1)$, the two latter equal to 0 by default, since the declaration of `threads` omitted them. Thread blocks form a grid of blocks—the dimensions of grids are specified similarly to dimensions of blocks and are used to identify blocks within grids (using triples `grid.x, grid.y, grid.z`). Thread and block identifiers are accessible within the kernel code through variables `threadIdx` and `blockIdx`.

The number of threads executing the `mv_kernel` is fixed and implied by dimensions of `threads` and `grid` arrays—in our example it was assumed that the number of threads is equal to $n$ and that $n$ is divisible by $BLOCK\_SIZE$ (if not simple adjustment has to be made).

For GPU threads to access variables in the GPU's DRAM memory (`A_G, x_G, y_G` in our example) they must be first allocated there and then their values transferred from the host memory, both operations done using special provided functions before kernel invocation.

A simple kernel function for implementing matrix-vector product on the Tesla architecture is given as Algorithm 2 on page 288.

## III. Matrix-vector multiplication algorithms

### A. Dense matrices

The CUDA algorithm presented in the previous section assumes that the thread that executes it performs all calculations for a single row of matrix `A_G` (this agrees with the assumption on the number of executing threads). As a consequence each thread can perform write operations perfectly in parallel with other threads but the performance can be poor due to not optimal memory accesses. The entries of matrix `A_G` (stored columnwise) transported to cache but not belonging to the row assigned to the thread are not used by it. To make the algorithm efficient the entries have to be used by other threads. That is the point where the proper arrangement of threads into thread blocks makes the difference.

**Algorithm 2** CUDA version of a simple matrix-vector product algorithm

```
__global__ void mv_kernel ( float* A_G,
  float* x_G, float* y_G, int n)
{
  int j;
  int i = blockIdx.x * BLOCK_SIZE
       + threadIdx.x;
  float t = 0.0f;
  for (j = 0; j < n; j ++ ) {
    t += A_G[i + j*n] * x_G[j];
  }
  y[i] = t;
}
```

Such an arrangement is done in the implementation of the standard SGEMV BLAS routine provided in the CUDA SDK [4], `cublasSgemv`. Blocks of threads are 1-dimensional with 128 threads per block and additionally fast shared memory is used for storing parts of vector $x$.

In [5] certain improvements to the original `cublasSgemv` algorithm have been proposed, consisting of using a texture memory unit to access elements of matrix $A\_G$ and grouping threads into 2-dimensional blocks of a fixed size 16x16.

### B. Banded matrices

For special types of matrices (banded, sparse, multi-diagonal, etc.) special algorithms can be used to exploit possible performance improvements. For all types of matrices the main indication is to perform operations only on non-zero elements. Despite the obvious intuitive advantage of such an approach the practical improvements may depend upon such details as the number of zeros in a matrix or influence of the memory access pattern on the performance.

In this and the next section two types of matrices frequently used in scientific computing are considered, banded matrices and sparse matrices stored in CRS (compressed row storage) format.

As an algorithm for banded matrices the implementation of SGBMV BLAS in CUDA SDK has been considered. The algorithm is very similar to the implementation of SGEMV in `cublasSgemv`. The main difference lies in accessing the elements of matrix `A_G`. As required by BLAS standards the matrix is stored columnwise, but with diagonals stored in rows. Hence, the matrix-vector product for the matrix `A_B` that stores entries of `A_G` in the required format and for a single thread that performs operations on a single row is shown, with some simplifications, as Algorithm 3 on page 289.

In the algorithm, `b` denotes bandwidth. `b` is the most important parameter determining the performance. For small values of bandwidth different threads still can use entries of `A_G` from cache but the values of `x_G` are different. The complex addressing of `A_G` may also pose problems for processing cores, since, unlike standard superscalar CPUs,

**Algorithm 3** Simple CUDA matrix-vector product algorithm for banded matrices

```
__global__ void
mvb_kernel ( float* A_G, float* x_G,
             float* y_G, int n, int b) {
  int j;
  int i = blockIdx.x * BLOCK_SIZE
        + threadIdx.x;
  int begin = max(0, i-b);
  int end = min(n, i+b+1)
  float t = 0.0f;
  for (j = begin; j < end; j++ ) {
    t += A_G[i+j*(2*b+1)-j+b] * x_G[j];
  }
  y[i] = t;
}
```

**Algorithm 4** Simple matrix-vector product algorithm for CRS storage format

```
void mv_crs( float* VA, int* JA, int* IA,
             int n, float* x, float* y ) {
  for(int i = 0; i < n; i++) {
    int size = IA[i+1] - IA[i];
    float sdot = 0.0f;
    for( int j = 0; j < size; j++ )
      sdot += VA[j]*x[JA[j]];
    y[i] = sdot
  }
}
```

they cannot perform integer and floating point operations in parallel.

### C. Sparse matrices

Sparse matrices are the most common form of matrices arising in approximations of partial differential equations describing different scientific and engineering problems. Hence, the importance of providing an efficient implementation of matrix-vector product, on which many iterative methods for systems of linear equations, like e.g. conjugate gradient or GMRES, can be based.

The CRS format [6] stores all non-zero entries of a matrix in a single one-dimensional vector, say VA, together with two other vectors: one, JA, with the same dimension as VA for storing column indices of the corresponding entries of the matrix and the other, say IA, with the dimension equal to the number of rows, for storing the indices of the first entries of subsequent rows in the array VA. A simple implementation of matrix-vector multiplication for CRS storage is shown as Algorithm 4 on page 289.

A CUDA implementation of Algorithm 4 has been proposed in [7]. The algorithm is shown as Algorithm 5 on page 289.

**Algorithm 5** Simple CUDA matrix-vector product algorithm for CRS storage format

```
__global__ void
mv_crs_kernel(float* VA, int* JA, int* IA,
              int n, float* x, float* y ) {
  __shared__ float cache[BLOCK_SIZE];
  int begin = blockIdx.x * BLOCK_SIZE;
  int end = block_begin + BLOCK_SIZE;
  int row = begin + threadIdx.x;
  int col;
  if(row < n) cache[threadIdx.x] = x[row];
  __syncthreads();
  if(row < n) {
    int r_b = IA[row];
    int r_e = IA[row+1];
    float sum = 0.0f;
    float xj;
    for(col = r_b; col<r_e; col++ ) {
      int j = JA[col];
      if( j >= begin && j < end )
        xj = cache[j-begin];
      else
        xj = x[j];
      sum += VA[col] * xj;
    }
    y[row] = sum;
} }
```

Again, one thread performs operations on a single row and elements of vector x are cached for better performance (the operation __syncthreads performs fast synchronization of threads belonging to a single block).

We propose a modification to Algorithm 5 consisting in using texture memory for elements of VA and JA. In Algorithm 5 the proper elements of both arrays are first put in texture memory (variable texture in the code) and than used by the corresponding threads. The update phase of the algorithm looks then as follows:

```
for(col = r_b; col<r_e; col++ ) {
  int j = texture.y;
  if( j>=block_begin && j<block_end )
    xj = cache[j-block_begin];
  else
    xj = x[j];
  sum += texture.x * xj;
}
y[row] = sum;
```

## IV. Experiments

### A. Hardware set-up

We tested the discussed algorithms on two platfroms. The first was Intel Xeon E5335, 2.0GHZ, with NVIDIA Tesla C870

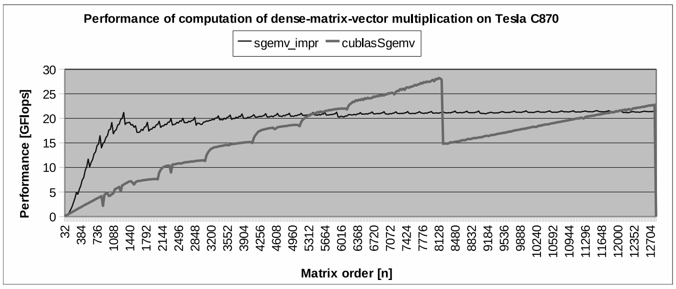


Fig. 2. Performance of matrix-vector multiplication algorithms for dense matrices on Tesla C870 (`cublasSgemv`, `sgemv_impr`)

as a CUDA device and ATI Radeon HD 2600 XT as a display device. The second was Intel Core 2 Duo E6750, 2.66 GHz, with ASUS EN8800GTX (with NVIDIA GeForce 8800GTX GPU as a CUDA and a display device). Both machines were running Fedora Core 8, 64-bit, with NVIDIA graphic driver version 177.13. The algorithms (CUDA and CUBLAS) presented previously, were implemented witch CUDA SDK version 2.0 beta. Note that, this version is the first that supports Fedora Core 8. For compilation we used gcc-compiler with standard optimization options (-*O*).

We tested the performance of matrix-vector product for three different matrix format representations: dense, sparse and banded, and measured the times for: allocating memory on graphic board, sending data from main memory to GPU memory, computing matrix-vector multiplication using card's DRAM and also getting results from GPU memory to main memory. All timing was performed by the functions provided by the CUDA environment.

### B. Performance

*1) Dense matrices:* We present first the results of two matrix-vector product algorithms for dense matrices residing already in GPU DRAM. The performance of `cublasSgemv` provided with CUDA SDK and the algorithm proposed in [5], denoted by `sgemv_impr`, is compared in Fig. 2 for matrices of dimension being a multiplicity of 32. Algorithm `sgemv_impr` presents more uniform performance and, moreover, does not exhibit a sudden break in performance for matrices of dimension not being the multiplicity of 32 (the effect reported in [5] which was also reproduced in our experiments). Nevertheless, the performance of any of the presented algorithms does not increase beyond 30 Gflops i.e. approx. 6% of the theoretical peak performance of the GPU (the theoretical limit derived from card's DRAM memory speed was between 37 and 50 Gflops).

For algorithm `sgemv_impr` we present also, in Figures 3–5, the timings for allocating data on a GPU and bandwidths for transporting data to and from GPU DRAM, both operations related to performing matrix-vector product on a GPU from a program running on a CPU (similar results were obtained for different types of matrices). The last figure, Fig. 6, shows the final performance of the matrix-vector operation performed on a GPU, taking into account the time necessary to transfer data to and from GPU.

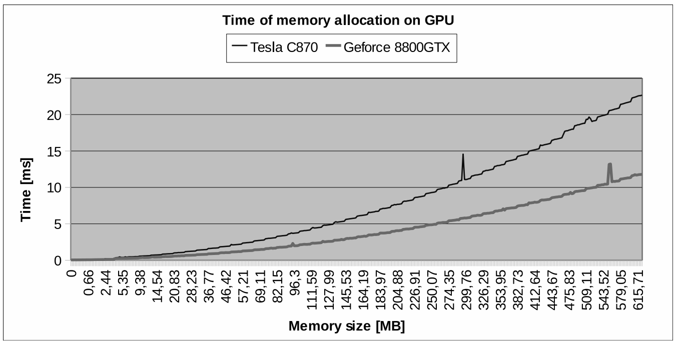


Fig. 3. Time of memory allocation on GPU for `sgemv_impr` algorithm and two CUDA platforms: Tesla C870 and GeForce8800GTX

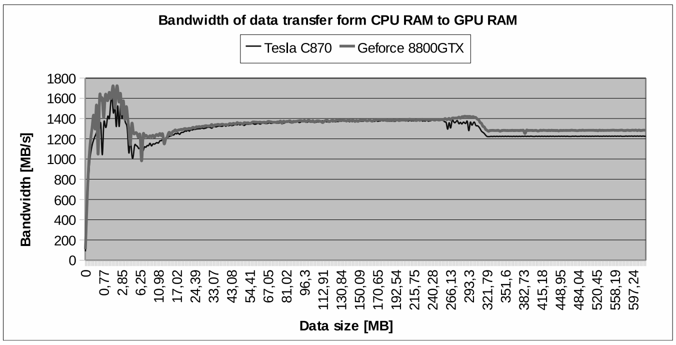


Fig. 4. Bandwidth of data transfer from CPU memory to GPU memory for `sgemv_impr` algorithm and two CUDA platforms: Tesla C870 and GeForce8800GTX

The timings and performance presented in Figures 3–6 show that the overall performance is determined almost exclusively by the time for transfering the entries of a matrix to GPU memory (for a matrix of size 10000 the transfer takes approx. 300 ms while all the other operations approx. 30 ms). The performance in Fig. 6 is slightly lower than that of a CPU (using the implementations from Intel MKL library we obtained 0.76 Gflops for Intel Xeon E5335 and 0.88 Gflops for Intel Xeon E5462). Hence, for scientific codes, especially iterative solvers employing matrix-vector product it is important how

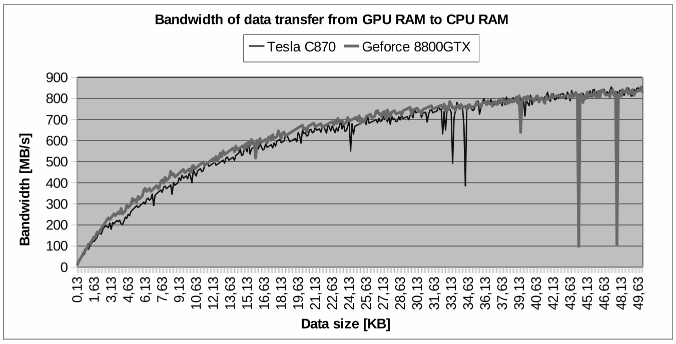


Fig. 5. Bandwidth of data transfer from GPU memory to CPU memory for `sgemv_impr` algorithm and two CUDA platforms: Tesla C870 and GeForce8800GTX

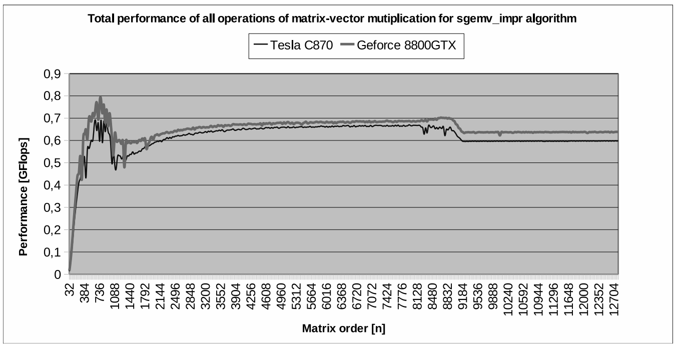


Fig. 6. Total performance of all operations of matrix-vector multiplication for `sgemv_impr` algorithm and two CUDA platforms: Tesla C870 and GeForce8800GTX

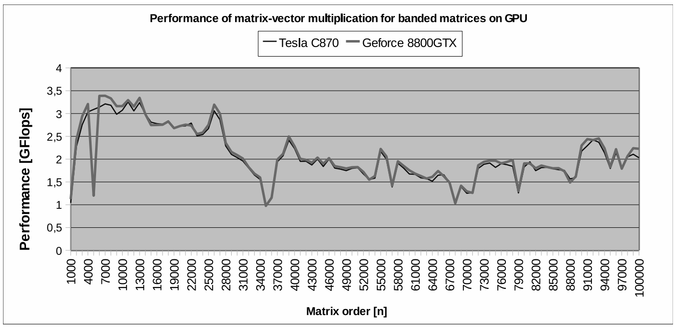


Fig. 7. Performance of matrix-vector multiplication for banded matrices, `cublasSgbmv` algorithm and bandwidth equal to 1.64% of matrix size (Tesla C870 and GeForce8800GTX)

many iterations will be performed on a given matrix, once it is transfered to the GPU memory.

*2) Banded matrices:* The results of experiments for banded matrices and the `cublasSgbmv` algorithm are presented in Fig. 7. The matrices had dimensions from 1000 to 100000 and the bandwidth was proportional to the matrix dimension (it was equal to 1.64% of matrix size to make the memory requirements for the largest matrices similar to memory requirements in dense matrix-vector product tests). A dramatic decrease in performance as compared to dense matrices can be observed. Moreover, the performance is far from uniform and strongly depends on the matrix size.

*3) Sparse matrices:* Two algorithms were tested in matrix-vector multiplication for sparse matrices stored in CRS format: the algorithm proposed in [7] ( Algorithm 5 on page 289) and its modification proposed in the current paper. The setting was similar to the setting for banded matrices, the dimensions of matrices changed from 1000 to 100000, but now the number of non-zero entries was equal to 1.64% of the matrix size. The results are presented in Fig. 8. Once again, a decrease in performance occured. The performance diminished with the increasing matrix size, as a result of more and more cache misses, since the non-zero entries of matrices occupied arbitrary positions in rows.

## V. Conclusion

Tesla architecture is still a new and evolving concept. For certain applications it can bring speed ups in the range of tens as compared to CPUs [3]. For the considered in the current paper problem of performing matrix-vector multiplication, the performance of cards with processors implementing the Tesla architecture vary significantly depending on the form of matrices as well as the overall context of calculations (e.g. whether the time necessary for transferring a matrix to GPU memory is taken into account or not). Recent experiences show that algorithms proposed for matrix-vector product, e.g. in CUDA SDK, still can be improved. One of slight modifications to published algorithms for sparse matrices in CRS format has been presented in the paper. For some matrix sizes it brings performance increase. We plan to investigate it further and report on the results in forthcoming papers.

## Acknowledgment

The authors would like to thank Mr Stanisław Rejowski and Mr Jacek Szwarga from the Action Company for providing the hardware on which the experiments were performed.

## References

[1] R. Strzodka, M. Doggett, and A. Kolb, "Scientific computation for simulations on programmable graphics hardware." *Simulation Modelling Practice and Theory, Special Issue: Programmable Graphics Hardware*, vol. 13(8), 2005, pp. 667–680.

[2] V. Volkov, and J. W. Demmel, "LU, QR and Cholesky factorizations using vector capabilities of GPUs", *Technical Report No. UCB/EECS-2008-49, EECS Department,University of California, Berkeley*, May 13, 2008.

[3] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, "NVIDIA Tesla: A Unified Graphics and Computing Architecture," *IEEE Micro,* vol. 28, 2008, pp. 39–55.

[4] NVIDIA CUDA 2008. CUBLAS Library 1.1 Guide http://developer.download.nvidia.com/compute/cuda/1_1/CUBLAS_Library_1.1.pdf

[5] N. Fujimoto, "Faster Matrix-Vector Multiplication on GeForce 8800GTX", In: *The proceedings of IEEE International Parallel & Distributed Processing (IPDPS) 2008*.

[6] B. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, " *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*", SIAM, Philadelphia, PA, 1994.

[7] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable Parallel Programming with CUDA" , *ACM Queue*, vol. 6, 2008

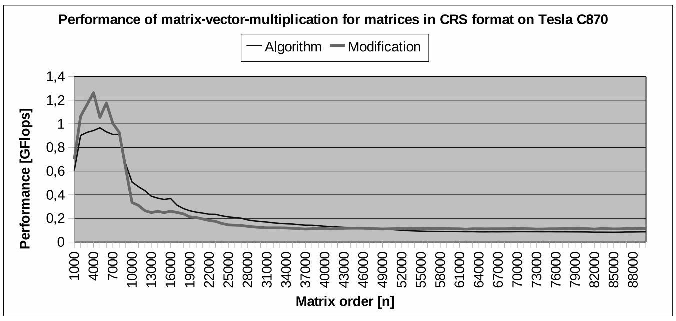


Fig. 8. Comparison of performance of two algorithms for sparse matrices in the CRS format: the algorithm proposed in [7] and its modification proposed in the current paper

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 293–296

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Solving a Kind of BVP for Second-Order ODEs Using Novel Data Formats for Dense Matrices

Przemysław Stpiczyński
Department of Computer Science, Maria Curie–Skłodowska University
Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland
Email: przem@hektor.umcs.lublin.pl

***Abstract*—The aim of this paper is to show that a kind of boundary value problem for second-order ordinary differential equations which reduces to the problem of solving tridiagonal systems of linear equations can be efficiently solved on modern multicore computer architectures. A new method for solving such tridiagonal systems of linear equations, based on recently developed algorithms for solving linear recurrence systems with constant coefficients, can be easily vectorized and parallelized. Further improvements can be achieved when novel data formats for dense matrices are used.**

## I. Introduction

LET consider the following boundary value problem which arises in many practical applications [10]:

$$-\frac{d^2u}{dx^2} = f(x) \quad \forall x \in [0,1] \tag{1}$$

where

$$u'(0) = 0, \quad u(1) = 0. \tag{2}$$

Numerical solution to the problem (1)–(2) reduces to the problem of solving tridiagonal systems of linear equations. Simple algorithms based on Gaussian elimination achieve poor performance, since they do not fully utilize the underlying hardware, i.e. memory hierarchies, vector extensions and multiple processors, what is essential in case of modern multicore computer architectures [1]. More sophisticated algorithms like *cyclic reduction*, *recursive doubling* [3] and *Wang's method* [15] lead to a substantial increase in the number of floating-point operations and do not utilize cache memory, what is crucial for achieving reasonable performance of parallel computers with a limited number of processors or modern multicore systems [4], [8], [14]. Following this observation, we have introduced a new fully vectorized version of the Wang's method devoted for solving linear recurrences with constant coefficients [12], [13] using two-dimensional arrays. In this paper we show how to apply this algorithm for solving tridiagonal systems of linear equations which arise after discretization of (1)–(2) and how to improve its performance using novel data formats for dense matrices [6].

## II. Simple solution

We want to find an approximation of the solution to the problem (1)–(2) in the following grid points

$$0 = x_1 < x_2 < \ldots < x_{n+1} = 1,$$

where $x_i = (i-1)h$, $h = 1/n$, $i = 1, \ldots, n+1$. Let $f_i = f(x_i)$ and $u_i = u(x_i)$. Using the following approximation

$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \tag{3}$$

we get

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i \tag{4}$$

for $i = 2, \ldots, n$. When we apply the boundary condition $u_{n+1} = u(1) = 0$, we get a special case of (4)

$$-u_{n-1} + 2u_n = h^2 f_n. \tag{5}$$

To find an approximation of $u''(0)$ using the boundary condition $u'(0) = 0$, let us observe [5] that

$$u''(0) \approx \frac{u(0-h) - 2u_1 + u_2}{h^2} \tag{6}$$

and

$$u'(0) \approx \frac{u_2 - u(0-h)}{2h}. \tag{7}$$

From (7) and $u'(0) = 0$, we get $u(0-h) \approx u_2$, thus using (1) and (6) we have

$$u_1 - u_2 = \frac{1}{2} h^2 f_1. \tag{8}$$

Finally, using (8), (4) and (5) respectively, we get the following system of linear equations [10]:

$$A\mathbf{u} = \mathbf{d} \tag{9}$$

where

$$A = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \tag{10}$$

and

$$\mathbf{u} = (u_1, \ldots, u_n)^T, \quad \mathbf{d} = (d_1, \ldots, d_n)^T \tag{11}$$

with $d_1 = \frac{1}{2}h^2 f_1$ and $d_i = h^2 f_i$, $i = 2, \ldots, n$. The solution to the system (9) can be found using Gaussian elimination without pivoting. The matrix $A$ defined by (10) can be

978-83-60810-14-9/08/$25.00 © 2008 IEEE

factorized as $A = LR$, where $L$ and $R$ are bidiagonal Toeplitz matrices

$$L = \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{pmatrix} \tag{12}$$

and

$$R = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & & 1 \end{pmatrix}. \tag{13}$$

Thus we have to solve two systems of linear equations

$$L\,(y_1,\dots,y_n)^T = \mathbf{d} \tag{14}$$

and then

$$R\mathbf{u} = (y_1,\dots,y_n)^T. \tag{15}$$

The first stage (forward reduction) of the algorithm for solving (9), namely finding the solution to the system (14), can be done using

$$\begin{cases} y_1 = d_1 \\ y_i = d_i + y_{i-1} \quad \text{for } i = 2,\dots,n. \end{cases} \tag{16}$$

Then (second stage, back substitution) we find the solution to the system (15) using

$$\begin{cases} u_n = y_n \\ u_i = y_i + u_{i+1} \quad \text{for } i = n-1, n-2, \dots, 1. \end{cases} \tag{17}$$

Note that there is no need to allocate an array for the elements $y_i$, $i = 1,\dots,n$. Assuming that initially the array u contains the elements of the vector $\mathbf{d}$, namely u(i)$= d_i$, the algorithm based on (16) and (17) can be expressed as the following simple loops

```
do i=2,n
  u(i)=u(i)+u(i-1)
end do
do i=n-1,1,-1
  u(i)=u(i)+u(i+1)
end do
```

Unfortunately, this simple algorithm cannot be vectorized or parallelized, so it can utilize only a small fraction of the theoretical peak performance of modern multicore multiprocessors.

## III. "Divide and conquer" approach

It is clear that (16) and (17) are the special cases of the following more general problem of solving *linear recurrence systems with constant coefficients*

$$y_k = \begin{cases} 0 & \text{for } k \le 0 \\ d_k + \sum\limits_{j=1}^{m} a_j y_{k-j} & \text{for } 1 \le k \le n, \end{cases} \tag{18}$$

where simply $m = 1$, $a_1 = 1$ and in case of (17), the unknowns are calculated from $n$ down to 1. In [12] we introduced a new algorithm based on Level 2 and 3 BLAS routines [3], which is a generalization of our earlier fully vectorized algorithm [13]. It can be efficiently implemented on various shared-memory parallel computers [11]. In this section we present a very brief description of the algorithm applied for $m = 1$, $a_1 = 1$ and show how to improve its performance using novel data formats for dense matrices.

### A. Simple 2D-array algorithm

The main idea of the algorithm is to rewrite (18) as a block-bidiagonal system of linear equations [12], [13]. Without loss of generality, let us assume that there exist two positive integers $r$ and $s$ such that $rs \le n$ and $s > 1$. The method can be used for finding $y_1,\dots,y_{rs}$. To find $y_{rs+1},\dots,y_n$, we use (16) directly. Then for $j = 1,\dots,r$, we define vectors

$$\mathbf{d}_j = \left(d_{(j-1)s+1},\dots,d_{js}\right), \quad \mathbf{y}_j = \left(y_{(j-1)s+1},\dots,y_{js}\right) \in \mathbb{R}^s$$

and find all $\mathbf{y}_j$ using the following formula

$$\begin{cases} \mathbf{y}_1 = L_s^{-1}\mathbf{d}_1 \\ \mathbf{y}_j = L_s^{-1}\mathbf{d}_j + y_{(j-1)s}\mathbf{e} \quad \text{for } j = 2,\dots,r \end{cases} \tag{19}$$

where $\mathbf{e} = (1,\dots,1)^T$ and the matrix $L_s \in \mathbb{R}^{s\times s}$ is of the same form as $L$ given by (12). However, it is better to find the matrix

$$Y = (\mathbf{y}_1,\dots,\mathbf{y}_r) \in \mathbb{R}^{s\times r}$$

instead of individual vectors $\mathbf{y}_j$, $j = 1,\dots,r$. Indeed, for $k = 2,\dots,s$ we perform

$$Y_{k,1:r} \leftarrow Y_{k,1:r} + Y_{k-1,1:r}, \tag{20}$$

and this operation (simplified AXPY [3]) can be vectorized and parallelized. Then we use (19) to find the last entry of each vector $\mathbf{y}_j$, $j = 2,\dots,r$, and finally we use (19) to find $s-1$ first entries of these vectors (note that this operation can also be vectorized and parallelized). This approach has one disadvantage: the number of floating-point operations required by the algorithm is twice as many as for the simple algorithm based on (16) (see [12]). Analogously, we can find the solution to (15) using

$$\begin{cases} \mathbf{u}_n = R_s^{-1}\mathbf{y}_r \\ \mathbf{u}_j = R_s^{-1}\mathbf{y}_j + u_{js+1}\mathbf{e} \quad \text{for } j = r-1,\dots,1, \end{cases} \tag{21}$$

where

$$\mathbf{u}_j = \left(u_{(j-1)s+1},\dots,u_{js}\right) \in \mathbb{R}^s$$

and $R_s \in \mathbb{R}^{s\times s}$ is of the same form as $R$ given by (13). Note that there is no need to allocate an array for the matrix $Y$. We can allocate a two-dimensional array u as the storage for the matrix

$$U = \begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & & \vdots \\ u_{s1} & \dots & u_{sr} \end{pmatrix} \in \mathbb{R}^{s\times r}. \tag{22}$$

and assign u(i,j)=$d_{(i-1)s+j}$, $i = 1,\dots,s$, $j = 1,\dots,r$. The The leading dimension of the array (LDA for short) can be LDA $= s$, however the performance of the algorithm can be improved by the use of *leading dimension padding* [9], what

```
        1 10 19 28 37 46 55 64 73 82
        2 11 20 29 38 47 56 65 74 83
        3 12 21 30 39 48 57 66 75 84
        4 13 22 31 40 49 58 67 76 85
    A = 5 14 23 32 41 50 59 68 77 86
        6 15 24 33 42 51 60 69 78 87
        7 15 25 34 43 52 61 70 79 88
        8 17 26 35 44 53 62 71 80 89
        *  *  *  *  *  *  *  *  *  *
```

Fig. 1. Standard column major storage of a $8 \times 10$ array with LDA=9.

```
         1  5  9 13 | 33 37 41 45 | 65 69  *  *
         2  6 10 14 | 34 38 42 46 | 66 70  *  *
         3  7 11 15 | 35 39 43 47 | 67 71  *  *
         4  8 12 16 | 36 40 44 48 | 68 72  *  *
    A = ---------------------------------------
        17 21 25 29 | 49 53 57 61 | 73 77  *  *
        18 22 26 30 | 50 54 58 62 | 74 78  *  *
        19 23 27 31 | 51 55 59 63 | 75 79  *  *
         *  *  *  * |  *  *  *  * |  *  *  *  *
```

Fig. 2. Square blocked full column major order of a $7 \times 10$ matrix with $4 \times 4$ blocks.

means that we insert unused array elements between columns (Fortran) or rows (C/C++) of a multidimensional array by increasing the leading dimension of the array (Figure 1). Usually, it is sufficient to set $\mathtt{LDA} = M$, where $M \geq s$ and $M$ is *not* a multiple of a power of two.

The algorithm for solving (9) can be easily parallelized. Each processor can be responsible for computing a block of columns of the matrix $U$. The first stage (forward reduction) proceeds as follows. During the first (parallel) step, for $k = 2, \ldots, s$, each processor applies the operation (20) restricted to its own columns. Then the last entries of all vectors $\mathbf{y}_j$ are calculated sequantially. Finaly, each processor uses (19) to find $s-1$ first entries of its columns. Similarly we can parallelize the second stage of the algoritm (back substitution).

The value of the parameter $r$ should be at least $r = vp$, where $p$ is the number of processors and $v$ is the length of vector registers in a vector processor (if applicable). However, the best performance is achieved if $r = O(\sqrt{n})$ and $s = O(\sqrt{n})$ [12]. It should be noticed that sometimes the performance of the algorithm can be unsatisfactory. When the standard column major storage is used, successive elements of vectors $Z_{k,*}$ and $Z_{k-1,*}$ in (20) do not occupy contiguous memory locations, thus may not belong to the same *cache line* and *cache misses* may occur [9].

### B. Using novel data formats

Suppose that we have a $m \times n$ matrix $A$. The matrix can be partitioned as follows

$$A = \begin{pmatrix} A_{11} & \ldots & A_{1n_g} \\ \vdots & & \vdots \\ A_{m_g1} & \ldots & A_{m_gn_g} \end{pmatrix}. \qquad (23)$$

Each block $A_{ij}$ contains a submatrix of $A$ and it is stored as a square $n_b \times n_b$ block which occupies a contiguous block of memory (Figure 2). The value of the blocksize $n_b$ should be chosen to map nicely into L1 cache [6], [7]. Note that when $n_g n_b \neq n$ or $m_g n_b \neq m$, then $A_{1n_g}, \ldots, A_{m_gn_g}$ or $A_{m_g1}, \ldots, A_{m_gn_g}$ are not square blocks and some memory will be wasted ('`*`' in Figure 2). As mentioned in [6] the novel data format can be represented by using a four dimensional array where `A(i,j,k,l)` refers to the `(i,j)` element within the `(k,l)` block.

Let us observe that the algorithm introduced in the previous subsection can be implemented using a matrix representation based on the *square blocked full data format* presented above. The matrix $U$ can be partitioned as (23). Then the algorithm will operate on small $n_b$-vectors within $(n_b \times n_b)$-blocks and the number of cache misses will be substantially reduced.

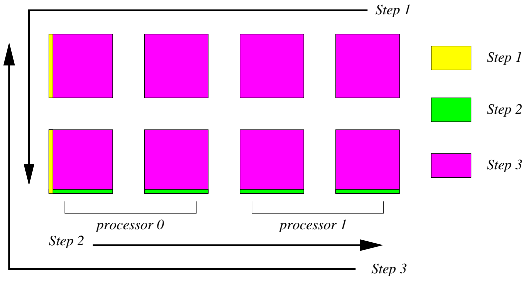


Fig. 3. The first stage using square blocked data format.

The first and the third step of the first stage can be done in parallel, while the second is sequential. To reduce the number of cache misses, blocks should be computed in the appropriate order. During the first step, columns of blocks should be computed from right to left and 'top-down' within each block column. Then during the second step we update elements in the lower row of blocks and finally (the third step) we update blocks from right to left and 'bottom-up' within each column of blocks (see Figure 3).

## IV. Results of experiments

Let us consider the algorithms described in the previous sections:

SEQ sequential algorithm based on (16) and (17),

DC divide and conquer algorithm based on (19) and (21) using two dimensional arrays,

ND divide and conquer algorithm based on (19) and (21) using the square blocked full data format.

The algorithms have been implemented in Fortran 95 with OpenMP directives [2] and all vector operations have been implemented using array section assignments. The experiments have been carried out on a dual processor Quad-Core Xeon (2.33 GHz, 12 MB L2 cache, 4 GB RAM running under Linux) workstation using the g95 Fortran Compiler.

We have measured the execution time (in seconds), the performance (in megaflops) of the algorithms for various values of $n$, $r$, $s$ and blocksizes $n_b$ and the speedup relative to Algorithm SEQ for optimal $n_b$, $r$, $s$ and various numbers of processors to find out how we can improve the performance of the simple scalar code (Algorithm SEQ) by the use of advanced computer hardware. Exemplary results are presented in Figures 4, 5 and Table I. The results of experiments can be summarized as follows.

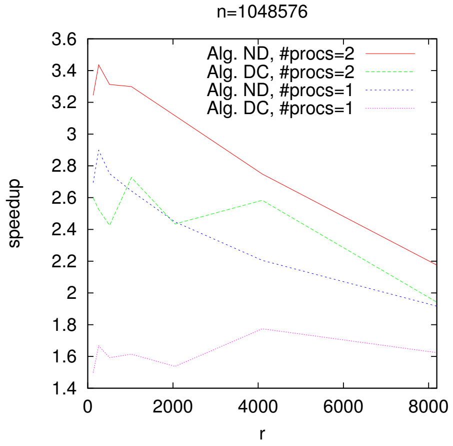


Fig. 4. Speedup of the algorithms DC and ND relative to Algorithm SEQ for various $r$.

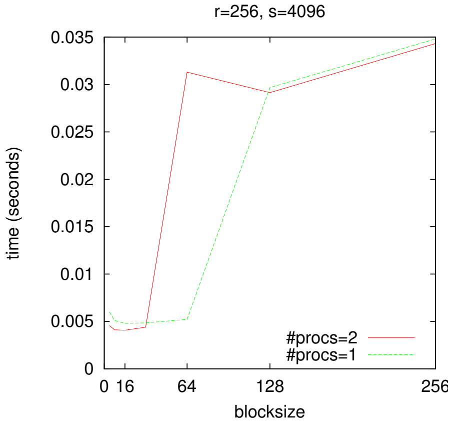


Fig. 5. Execution time of Algorithm ND for various blocksizes $n_b$.

TABLE I
EXECUTION TIME, SPEEDUP AND PERFORMANCE OF THE ALGORITHMS SEQ, DC, ND FOR $n = 1048576$

| | Time | | Speedup | | Mflops | |
|---|---|---|---|---|---|---|
| alg. | #p=1 | #p=2 | #p=1 | #p=2 | #p=1 | #p=2 |
| SEQ | 1.40E-2 | 1.40E-2 | – | – | 225 | 225 |
| DC | 7.96E-3 | 5.49E-3 | 1.75 | 2.55 | 659 | 954 |
| ND | 4.80E-3 | 4.06E-3 | 2.89 | 3.43 | 1093 | 1286 |

Both introduced algorithms (DC and ND) run faster than Algorithm SEQ. The performance of Algorithm ND depends on the blocksize $n_b$, so it should be chosen carefully. The optimal blocksize is $n_b = 16$ (Figure 5). When the blocksize $n_b$ is too large, the performance of the algorithm decreases dramatically because blocks do not fit into L1 cache and cache misses occur. For the optimal blocksize, the algorithm achieves reasonable speedup relative to Algorithm SEQ. Moreover, in case of the simple algorithm, fast vector instructions cannot be used without manual optimization of the simple scalar code. Although all optimization switches have been turned on, the compiler has produced rather slow output for the scalar code. The values of the parameters $r$ and $s$ should be chosen to minimize the number of flops required by the algorithms, however $r = s$ is rather a good choice in case of Algorithm DC and $r = \sqrt{n}/4$ in case of Algorithm ND. Finally, comparing the performance of the new algorithm which utilizes the novel data format with the performance of the divide and conquer algorithm which uses standard Fortran two-dimensional arrays with the column major storage order, we can observe that Algorithm ND is faster than Algorithm DC.

## V. CONCLUSION

We have shown that the performance of the simple algorithm for solving a kind of boundary value problem for second-order ordinary differential equations which reduces to the problem of solving tridiagonal systems of linear equations can be highly improved by the use of the divide and conquer vectorized algorithm which operates on the square blocked full data format for dense matrices. The algorithm can also be parallelized, thus it should be suitable for novel multicore architectures.

## REFERENCES

[1] A. Buttari, J. Dongarra, J. Kurzak, J. Langou, P. Luszczek, and S. Tomov, "The impact of multicore on math software," *Lecture Notes in Computer Science*, vol. 4699, pp. 1–10, 2007.

[2] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon, *Parallel Programming in OpenMP*. San Francisco: Morgan Kaufmann Publishers, 2001.

[3] J. Dongarra, I. Duff, D. Sorensen, and H. Van der Vorst, *Numerical Linear Algebra for High Performance Computers*. Philadelphia: SIAM, 1998.

[4] E. Elmroth, F. Gustavson, I. Jonsson, and B. Kågström, "Recursive blocked algorithms and hybrid data structures for dense matrix library software," *SIAM Rev.*, vol. 46, pp. 3–45, 2004.

[5] G. Golub and J. M. Ortega, *Scientific Computing: An Introduction with Parallel Computing*. Academic Press, 1993.

[6] F. G. Gustavson, "New generalized data structures for matrices lead to a variety of high performance algorithms," *Lect. Notes Comput. Sci.*, vol. 2328, pp. 418–436, 2002.

[7] ——, "High-performance linear algebra algorithms using new generalized data structures for matrices," *IBM J. Res. Dev.*, vol. 47, pp. 31–56, 2003.

[8] ——, "The relevance of new data structure approaches for dense linear algebra in the new multi-core / many core environments," *Lecture Notes in Computer Science*, vol. 4967, pp. 618–621, 2008.

[9] M. Kowarschik and C. Weiss, "An overview of cache optimization techniques and cache-aware numerical algorithms," *Lecture Notes in Computer Science*, vol. 2625, pp. 213–232, 2003.

[10] L. R. Scott, T. Clark, and B. Bagheri, *Scientific Parallel Computing*. Princeton University Press, 2005.

[11] P. Stpiczyński, "Numerical evaluation of linear recurrences on various parallel computers," in *Proceedings of Aplimat 2004, 3rd International Conference, Bratislava, Slovakia, February 4–6, 2004*, M. Kovacova, Ed. Technical Univerity of Bratislava, 2004, pp. 889–894.

[12] ——, "Solving linear recurrence systems using level 2 and 3 BLAS routines," *Lecture Notes in Computer Science*, vol. 3019, pp. 1059–1066, 2004.

[13] P. Stpiczyński and M. Paprzycki, "Fully vectorized solver for linear recurrence systems with constant coefficients," in *Proceedings of VECPAR 2000 – 4th International Meeting on Vector and Parallel Processing, Porto, June 2000*. Facultade de Engerharia do Universidade do Porto, 2000, pp. 541–551.

[14] P. Stpiczyński, "Evaluating linear recursive filters using novel data formats for dense matrices," *Lecture Notes in Computer Science*, vol. 4967, pp. 688–697, 2008.

[15] H. Wang, "A parallel method for tridiagonal equations." *ACM Trans. Math. Softw.*, vol. 7, pp. 170–183, 1981.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 297–301

ISBN 978-83-60810-14-9
ISSN 1896-7094

# On the hidden discrete logarithm for some polynomial stream ciphers

Vasyl Ustimenko
University of Maria Curie Sklodowska
pl. Maria Curie Sklodowskiej 1, 20-031 Lublin, Poland
Email: vasyl@hektor.umcs.lublin.pl

***Abstract*—The paper is devoted to the special key management algorithm for the stream ciphers defined in [12] via finite automata corresponding to the family of directed algebraic graphs of high girth and two affine transformation. Security of the key based on the complexity of the discrete logarithm problem. It has additional heuristic security because of the "hidden base" and "hidden value" of the discrete logarithm function. We consider the heuristic evaluation of the complexity for different attack by the adversary on our private key cipher. The detailed description of the whole algorithm is given. Implemented software package has been used for the evaluation of mixing properties and speed of the private key encryption.**

## I. Introduction

WE WILL introduce the key management block for the stream ciphers defined in [12] via finite automata corresponding to the family of directed algebraic graphs of high girth. The time evaluation of software package implementing these algorithms compares well with the performance of fast but not very secure RC4, DES, algorithms based on simple graphs (symmetric anti reflexive binary relations) developed during last ten years [4].

In Section 3 we described modified algorithm in term of arithmetical dynamical systems. We add the key management block to our algorithm and consider the heuristic evaluation of active attacks by the adversary.

The explicit description of the dynamical system is given in last section in terms of corresponding finite automata. We give the explicit formula for invariants we use in algorithm.

Section 4 devoted to evaluation of speed of the encryption in case of special rings of kind $Z_2^s$, $s \geq 8$.

## II. Basic Cryptographical Terminology

Assume that an unencrypted message, *plaintext*, which can be image data, is a string of bytes. It is to be transformed into an encrypted string or *ciphertext*, by means of a cryptographic algorithm and a *key*: so that the recipient can read the message, encryption must be *invertible*.

Conventional wisdom holds that in order to defy easy decryption, a cryptographic algorithm should produce seeming chaos: that is, ciphertext should look and test random. In theory an eavesdropper should not be able to determine any significant information from an intercepted ciphertext. Broadly speaking, attacks to a cryptosystem fall into 2 categories: *passive attacks*, in which adversary monitors the communication channel and *active attacks*, in which the adversary may transmit messages to obtain information (e.g. ciphertext of chosen plaintext).

An assumption first codified by Kerckhoffs in the nineteen century is that the algorithm is known and the security of algorithm rests entirely on the security of the key.

Cryptographers have been improving their algorithms to resist the following two major types of attacks:

(i) *ciphertext only*—the adversary has access to the encrypted communications.

(ii) *known plaintext*—the adversary has some plaintextx and corresponding ciphertexts.

Nowadays the security of the plaintext rests on encryption algorithm (or private key algorithm), depended on chosen key (password), which has good resistance to attacks of type (i) and (ii), and algorithm for the key exchange (public keys) with good resistance to active attacks, when the adversary can generate each plaintext $p$ and get the corresponding plaintext $c$ (see [2], [3], [9] or [10]). The combination of appropriate private key and public key algorithms can lead to symmetric algorithm with good resistance even to active attacks. The example of such combination will be given in the next section of the paper.

## III. Arithmetical Dynamical Systems as Encryption Tools

Let $K$ be the commutative ring, $F(K) = K[t, x_1, x_2, \dots]$ is the ring of all polynomials in variables $t, x_1, x_2, \dots$. We use symbol $\mathrm{Reg}(K)$ for the totality of regular elements i.e not a zero divisors: $a \in \mathrm{Reg}(K)$ implies $a \times x \neq 0$ for each $x \neq 0$. Let $K^\infty = \{\mathrm{x} = (t, x_1, x_2, \dots)|x_i \in K, t \in K, \mathrm{supp}(\mathrm{x}), \infty\}$ and $K^n = \{(x_1, x_2, \dots, x_n)|x_i \in K\}$.

Let us consider two polynomial maps $P$ and $R$ of $K^\infty$ into $K^\infty$:

$$(t, x_1, x_2, \dots, ) \to (t, P_1(t, x_1, x_2, \dots), P_2(t, x_1, x_2, \dots), \dots)$$

and

$$(t, x_1, x_2, \dots, ) \to (t, R_1(t, x_1, x_2, \dots), \\ R_2(t, x_1, x_2, \dots), \dots),$$

where $P_i(t, x_1, x_2, \dots)$ and $R_i(t, x_1, x_2, \dots)$, $i = 1, 2, \dots$ are elements of $F(K)$.

We consider two families: $f_t^n$ and ${g_t}^n$ of $K^n$ onto $K^n$ sending $(x_1, x_2, \dots, x_n)$ to

$$(P_1'(t, x_1, x_2, \dots), P_2'(t, x_1, x_2, \dots), P_n'(t, x_1, x_2, \dots x_n))$$

and

$$(R'_1(t, x_1, x_2, \dots), R'_2(t, x_1, x_2, \dots), R'_n(t, x_1, x_2, \dots x_n)),$$

where $P'_i$ and $R'_i$, $i = 1, 2, \dots, n$ correspond to the specialisations $x_{n+1} = 0, x_{n+2} = 0, \dots$ of $P_i$ and $R_i$ associated with the pair ($P$, $R$). We identify $f_t$ and $g_t$, $t \in K$ with the corresponding maps $K^n \to K^n$

Let $\mathrm{r} = (r_1, r_2, \dots r_t) \in \mathrm{Reg}(K)^t$ be the tuple of length $l(\mathrm{r})) = t$. We introduce $F_{\mathrm{r}}$, as the composition of maps $f_{r_1}, g_{r_2}, \dots, f_{r_{2s-1}}, g_{r_{2s}}$ in case of $t = 2s$ and as composition of $f_{r_1}, g_{r_2}, \dots, f_{r_{2s-1}}, g_{r_{2s}}, f_{r_{2s+1}}$ for $t = 2s + 1$.

We say that the pair $P$ and $R$ form an arithmetical dynamical system depending on time $t$ if the following conditions hold

1) existence of $\mathrm{x} = (x_1, \dots, x_n) \in K^n$ such that $f_{t_1}(x_1, x_2, \dots, x_n) = f_{t_2}(x_1, x_2, \dots, x_n)$ for some $t_1$ and $t_2$ implies the equality $t_1 = t_2$.

2) maps $f_t$ and $g_t$, $t \in K$ are bijections and $f_{-t}$ and $g_{-t}$ are inverse maps for them.

3) There is a constant $c$, $c > o$ such that for each pair of tuples r, b of regular elements, conditions $l(\mathrm{r}) \leq cn$, $l(\mathrm{r}) \leq cn$ and $F_{\mathrm{r}}(\mathrm{x}) = F_{\mathrm{b}}(\mathrm{x})$ for some x implies $\mathrm{r} = \mathrm{b}$.

If $(P, R)$ form an arithmetical dynamical system, then the inverse of $F_{\mathrm{r}}$, $l(\mathrm{r}) = 2s + 1$ is $F_{\mathrm{b}}$, where

$$\mathrm{b} = (-r_{2s+1}, -r_{2s}, \dots, -r_1).$$

If $l(\mathrm{r}) = 2s$ then $F_{\mathrm{r}}^{-1}$ is the composition of $g_{-r_{2s}}$ and $F_{\mathrm{d}}$, where $\mathrm{d} = (-r_{2s-1}, -r_{2s-2}, \dots, -r_1)$.

We can treat $K^n$ as the plainspace, refer to the union U of $\mathrm{Reg}(K)^t$, $1 < t < cn$ as the key space and treat $\mathrm{x} \to F_{\mathrm{a}}(\mathrm{x})$ as the encryption map corresponding to the key a. The ciphertext $y = F_{\mathrm{a}}(\mathrm{x})$ can be decrypted by the map $F_{\mathrm{a}}^{-1}$ written above. So the algorithm is symmetrical. The property 3 implies that different keys of length $< cn$ produce distinct ciphertexts.

We introduce the following directed graph $\phi = \phi(n)$ corresponding to maps ${f_t}^n$ and ${g_t}^n$ over $K^n$. Firstly we consider two copies of $P$ (set of points) and $L$ (set of lines) of the free module $K^n$. We connect point $p \in P$ with the line $l \in L$ by directed arrow if and only if there is $t \in \mathrm{Reg}(K)$ such that $f_t(p) = l$. Let $t$ be the colour of such a directed arrow. Additionally we join $l \in L$ and $p \in P$ by directed arrow with the colour $t$ if there is $t \in \mathrm{Reg}(K)$ such that $g_t(l) = p$. We can consider $\phi$ as finite automaton for which all states are accepting states. We have to chose point $p$ (plaintext) as initial state. It is easy to see that $f_t$ and $g_t$ are the transition functions of our automaton. Let $t_1, \dots, t_s$ be the "program" i.e. sequence of colours from $\mathrm{Reg}(K)$. Then the computation is the directed pass $p$, $f_{t_1}(p) = p^1, g_{t_2}(p^1) = p^2, \dots$. If $s$ is even then the last vertex is $f_{t_s}(p^{s-1})$, in case of odd $s$ we get $g_{t_s}(p^{s-1}) = p^s$ as the result of the computation (encryption). The stop of the automata corresponds just to the absence of the next colour.

The inverse graph $\phi(n)^{-1}$ can be obtained by reversing of all arrows in $\phi$. We assume that colours of arrow in $\phi$ and its reverse in $\phi^{-1}$ are the same. So we can consider $\phi(n)^{-1}$ as an automaton as well. Then the decryption procedure starting from the ciphertext $p^s$ corresponds to the pass in $\phi^{-1}$ defined by sequence of colours $-t_s, -t_{s-1}, \dots, -t_1$.

Finally, we can consider well defined projective limit $\phi$ of automata $\phi^n$, $n \to \infty$ with the transition function $P_t(x_1, x_2, \dots) = P(t, x_1, x_2, \dots)$ and $R_t(x_1, x_2, \dots) = R(t, x_1, x_2, \dots)$. In case of finite $K$ we can use $\phi$ as a Turing machine working with the potentially infinite text in the alphabet $K$. Results of [41] allow to formulate the following statement.

THEOREM 1

*For each commutative ring $K$ there are a cubical polynomial maps $P$ and $R$ on $K^\infty$ forming arithmetical dynamical system with the constant $c \geq 1/2$ such that for each string* r *of elements from* $\mathrm{Reg}(K)$ *the polynomial map* $F_{\mathrm{r}}$ *is cubical.*

The example as above has been defined explicitly in [5] in graph theoretical terms. The maps $P$ and $R$ will stand further for that particular example. Corresponding to $(P, R)$ graphs $\phi(n)$ are strongly connected i.e. from the existence of directed pass from vertex $v$ to $w$ follows that $w$ and $v$ are connected by a directed pass. So connected components of $\phi(n)$ are well defined.

We combine the encryption process $F_{\mathrm{r}}$ corresponding to finite automaton $\phi(n)$ and string r of elements from $\mathrm{Reg}(K)$ with two invertible sparse affine transformation $\mathrm{Af}_1$ and $\mathrm{Af}_2$ and use the composition $\mathrm{Af}_1 \times F_{\mathrm{r}} \times \mathrm{Af}_2$ as encryption map. We refer to such a map as *deformation* of $F_{\mathrm{r}}$. In case of $\mathrm{Af}_1 = {\mathrm{Af}_2}^{-1}$ we use term *desynchronization*. In case of desynchronization the ciphertext is always distinct from the plaintext. We assume that $\mathrm{Af}_1$ and $\mathrm{Af}_2$ are parts of the key. Deformated or desynchronised encryption is much more secure, because it prevents adversary to use group automorphisms and special ordering of variables during his/her attacks.

In the case of deformation with fixed $\mathrm{Af}_1$ and $\mathrm{Af}_2$ and flexible r the property that the different passwords of kind r lead to different ciphertexts is preserved, but the situation, where the plaintext and corresponding ciphertext are the same can happen. Anyway the probability of such event is $1/|V|$, where $V = K^n$ is the plainspace.

### A. Watermarking Equivalence and Hidden Discrete Logarithm

THEOREM 2

*Let $\phi(n)$, $n \geq 6$ be the directed graph with the vertex set $K^{k+1}$ defined above for the pair $(P, R)$.*

*(i) There are the tuple $a = a(\mathrm{x})$, $\mathrm{x} \in K^{n+1}$ of quadratic polynomials $a_2, a_3, \dots, a_t$, $t = [(n+2)/4]$ in $K[x_0, x_1, \dots, x_n]$ such that for each directed pass $u = v_0 \to v_1 \to v_n = v$ we have $a(u) = a(v)$.*

*(ii) For any $t-1$ ring elements $x_i \in K)$, $2 \leq t \leq [(k+2)/4]$, there exists a vertex $v$ of $\phi(n)$ for which $a(v) = (x_2, \dots, x_t) = (x)$. So classes of equivalence relation $\tau = \{(u, v)|a(u) = a(v)\}$ are in one to one correspondents with the tuples in $K^t$.*

*(iii) The equivalence class $C$ for the equivalence relation $\tau$ on the set $K^{n+1} \cup K^{n+1}$ is isomorphic to the affine variety $K^t \cup K^t$ , $t = [4/3n]+1$ for $n = 0, 2, 3 \bmod 4$, $t = [4/3n]+2$ for $n = 1 \bmod 4$.*

We refer to $\tau$ as watermarking equivalence and call $C$ as above generalised connected component of the graph,

Let $|K| = d$ and $\eta$ numerating function i.e bijection between $K$ and $\{0, 1, \ldots, d-1\}$. For each tuple $\mathrm{t} = (t_0, t_1, \ldots t_s) \in K^{s+1}$ we consider its number $\eta(\mathrm{t}) = \eta(t_0) + \eta(t_1)d + \cdots + \eta(t_s)d^s$. Let $\mathrm{Reg}(K) = b \geq 2$, $\mu$ be the bijection between $\mathrm{Reg}(K)$ and $\{0, 1, \ldots, b-1\}$. We obtain $\mathrm{reg}(\mathrm{t})$ by taking the string of digits for $\eta(\mathrm{t})$ base $b$ and computing $\mu^{-1}$ for each digit. So $\mathrm{reg}(\mathrm{t})$ is a string of characters from the alphabet $\mathrm{Reg}(K)$.

THE ALGORITHM: Correspondents Alice and Bob are taking smallest prime number $p$ from interval $(b^{[(n+5)/2c]}, b^{[(n+5)/2]})$, where $c$ is some constant $> 3/2$ and some number $m$, $m < p$. Alice takes the plainspace $\mathrm{x}$ computes string $a(\mathrm{x})$ (see theorem ?), then $z = \eta(a(\mathrm{x}))$ and $u = z^m \bmod p$. She treats $u$ as integer and takes string $d(\mathrm{x}) = \mathrm{reg}(u)$ of characters from $\mathrm{Reg}(K)$. Her encryption is $\mathrm{Af}_1 \times F^n_{d(\mathrm{x}} \times \mathrm{Af}_2$. We think that numbers $m$, $c$ and fixed maps $\mathrm{Af}_i$, $i = 1, 2$ are parts of the key.

Let c be the ciphertext. Bob computes $z$ defined above az $z = \eta(a(\mathrm{c}))$, computes string $d(\mathrm{x})$ and use decryption map $(\mathrm{Af}_2)^{-1} \times {F^n_{d(\mathrm{x})}}^{-1} \times \mathrm{Af}_1$.

Let $C_n(\mathrm{x}) = C(\mathrm{x})$ be the encryption function corresponding to deformation of dynamical system. The adversary may try to find the factorization $C_n(\mathrm{x}) = ((\mathrm{Af}_1)) \times F^n_{d(\mathrm{x})} \times \mathrm{Af}_2$, where $Af_i, i = 1, 2$ are unknown and the function $d(\mathrm{x})$ is $\mathrm{reg}((\eta(a(\mathrm{x})^m))$, where $m$ is unknown also. During his active attack he can compute finite number of values $C(\mathrm{x}_i)$, $i \in J$ and use this information for finding the factorization. The following heuristic argument demonstrate that it can be difficult task.

Let as assume that affine transformation $\mathrm{Af}_i$, $i = 1, 2$ are known for adversary. Notice that finding them can be very difficult. Then the adversary can compute $\mathrm{d}_i = F^n_{d(\mathrm{x}_i)} = (\mathrm{Af}_1)^{-1} C(\mathrm{x}_i)(\mathrm{Af}_2)^{-1}$. The pass between vertices of the graph is unique. The Dijkstra algoritm is not suitable for finding the pass because the vertex space of the graph is the plainspace. But may be large group of automorphisms (see [14] and further referenes) will allow to find the pass. Then the adversary computes number $b_i = \eta(a(\mathrm{x})^m$ modulo known big prime. Still he is not able to find number $m$ because of the complexity of discrete logarithm problem. So he has to take for the set $\{\mathrm{x}_i | i \in J\}$ the totality of representatives from classes of watermarking equivalence (transversal). So $|J| > O(|K|^{[1/4]})$ because of theorem 2.

We use term *hidden discrete logarithm* for the name of modified algorithm because affine transformation do not allow the adversary to compute the class of watermarking equivalence containing the plaintext (base of the logarithm) and pass in the finite automaton corresponding to the value of the logarithm.

## IV. Explicit Construction, Algebraic Graphs of Arithmetical Dynamical System

Missing graph theoretical definitions the reader can find in [1] or [8]. E. Moore [7] used term *tactical configuration* of order $(s, t)$ for biregular bipartite simple graphs with bidegrees $s+1$ and $r+1$. It corresponds to incidence structure with the point set $P$, line set $L$ and symmetric incidence relation $I$. Its size can be computed as $|P|(s+1)$ or $|L|(t+1)$.

Let $F = \{(p, l) | p \in P, l \in L, pIl\}$ be the totality of flags for the tactical configuration with partition sets $P$ (point set) and $L$ (line set) and incidence relation $I$. We define the following irreflexive binary relation $\phi$ on the set $F$:

Let $(P, L, I)$ be the incidence structure corresponding to regular tactical configuration of order $t$.

Let $F_1 = \{(l, p) | l \in L, p \in P, lIp\}$ and $F_2 = \{[l, p] | l \in L, p \in P, lIp\}$ be two copies of the totality of flags for $(P, L, I)$. Brackets and parenthesis allow us to distinguish elements from $F_1$ and $F_2$. Let $DF(I)$ be the *double directed flag graph* on the disjoint union of $F_1$ with $F_2$ defined by the following rules

$(l_1, p_1) \to [l_2, p_2]$ if and only if $p_1 = p_2$ and $l_1 \neq l_2$,
$[l_2, p_2] \to (l_1, p_1)$ if and only if $l_1 = l_2$ and $p_1 \neq p_2$.

We will define below the family of graphs $D(k, K)$, where $k > 2$ is positive integer and $K$ is a commutative ring, such graphs have been considered in [5] for the case $K = F_q$.

let $P$ and $L$ be two copies of Cartesian power $K^N$, where $K$ is the commutative ring and $N$ is the set of positive integer numbers. Elements of $P$ will be called *points* and those of $L$ *lines*.

To distinguish points from lines we use parentheses and brackets: If $x \in V$, then $(x) \in P$ and $[x] \in L$. It will also be advantageous to adopt the notation for co-ordinates of points and lines introduced in [5] for the case of general commutative ring $K$:

$$(p) = (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p'_{2,2}, \ldots, p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,i}, \ldots)$$

$$[l] = [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l'_{2,2}, \ldots, l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,i}, \ldots]$$

The elements of $P$ and $L$ can be thought as infinite ordered tuples of elements from $K$, such that only finite number of components are different from zero.

We now define an incidence structure $(P, L, I)$ as follows. We say the point $(p)$ is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations between their co-ordinates hold:

$$\begin{aligned} l_{i,i} - p_{i,i} &= l_{1,0} p_{i-1,i} \\ l'_{i,i} - p'_{i,i} &= l_{i,i-1} p_{0,1} \\ l_{i,i+1} - p_{i,i+1} &= l_{i,i} p_{0,1} \\ l_{i+1,i} - p_{i+1,i} &= l_{1,0} p'_{i,i} \end{aligned} \tag{1}$$

(This four relations are defined for $i \geq 1$, $p'_{1,1} = p_{1,1}$, $l'_{1,1} = l_{1,1}$). This incidence structure $(P, L, I)$ we denote as $D(K)$. We identify it with the bipartite *incidence graph* of $(P, L, I)$, which has the vertex set $P \cup L$ and edge set consisting of all pairs $\{(p), [l]\}$ for which $(p)I[l]$.

For each positive integer $k \geq 2$ we obtain an incidence structure $(P_k, L_k, I_k)$ as follows. First, $P_k$ and $L_k$ are obtained from $P$ and $L$, respectively, by simply projecting each vector onto its $k$ initial coordinates with respect to the above order. The incidence $I_k$ is then defined by imposing the first $k-1$ incidence equations and ignoring all others. The incidence graph corresponding to the structure $(P_k, L_k, I_k)$ is denoted by $D(k, K)$.

The incidence relation motivated by the linear interpretation of Lie geometries in terms their Lie algebras. $\alpha$ belongs to the root system

$$\mathrm{Root} = \{(1,0),(0,1),(1,1),(1,2),(2,1),(2,2),(2,2)'\ldots, \\ (i,i),(i,i)',(i,i+1),(i+1,i)\ldots\}.$$

The “root system”

$$\mathrm{Root} = \{(1,0),(0,1),(1,1),(1,2),(2,1),(2,2), \\ (2,2)'\ldots,(i,i),(i,i)',(i,i+1),(i+1,i)\ldots\}$$

contains all real and imaginary roots of the Kac-Moody Lie Algebra $\tilde{A}_1$ with the symmetric Cartan matrix. We just doubling imaginary roots $(i,i)$ by introducing $(i,i)'$.

To facilitate notation in future results, it will be convenient for us to define $p_{-1,0} = l_{0,-1} = p_{1,0} = l_{0,1} = 0$, $p_{0,0} = l_{0,0} = -1$, $p'_{0,0} = l'_{0,0} = -1$, and to assume that (1) are defined for $i \geq 0$.

Notice that for $i = 0$, the four conditions (1) are satisfied by every point and line, and, for $i = 1$, the first two equations coincide and give $l_{1,1} - p_{1,1} = l_{1,0}p_{0,1}$.

Let $DE(n,K)$ ($DE(K)$) be the double directed graph of the bipartite graph $D(n,K)$ ($D(K)$, respectively). Remember, that we have the arc $e$ of kind $(l^1, p^1) \rightarrow [l^2, p^2]$ if and only if $p^1 = p^2$ and $l^1 \neq l^2$. Let us assume that the colour $\rho(e)$ of arc $e$ is $l^1_{1,0} - l^2_{1,0}$. Recall, that we have the arc $e'$ of kind $[l^2, p^2] \rightarrow (l^1, p^1)$ if and only if $l^1 = l^2$ and $p^1 \neq p^2$. let us assume that the colour $\rho(e')$ of arc $e'$ is $p^1_{1,0} - p^2_{1,0}$.

It is easy to see that the vertex set of the new graph is isomorphic to $K^{n+1} \cup K^{n+1}$. If $K$ is finite, then the cardinality of the colour set is $(|K| - 1)$. Let $\mathrm{Reg}K$ be the totality of regular elements, i.e. not zero divisors. Let us delete all arrows with colour, which is a zero divisor. New graph $RDE(t,K)$ ($RD(K)$) with the induced colouring is the automaton in the alphabet $\mathrm{Reg}(K)$.

Let $P_t(x_{1,0}, x_{0,1}, x_{11}, \ldots)$ and $R_t(x_{1,0}, x_{0,1}, x_{11}, \ldots)$ are the transition function of infinite graph $RD(K)$ of taking the neighbour of vertex from the first and second copy of the flag set for $D(K)$. The connected components of graph $D(n,K)$ can be given in the following way.

Finally we define the tuple $\mathrm{a}$ of theorem 2.

Graph $\phi(n)$ is the double flag graph for $D(k,K)$. We assume that $k \geq 6$and $t = [(n+2)/4]$. Each flag $f$ from $F_1 \cup F_2$ contains the unique point $u$ $u = (u_{01}, u_{11}, \cdots, u_{tt}, u'_{tt}, u_{t,t+1}, u_{t+1,t}, \cdots)$ of $\mathrm{D}(n,K)$ . For every $r$, $2 \leq r \leq t$, let

$$a_r(f) = a_r(u) = \sum_{i=0,r} (u_{ii}u'_{r-i,r-i} - u_{i,i+1}u_{r-i,r-i-1}),$$

and $a = a(u) = (a_2, a_3, \cdots, a_t)$. So in fact each polynomial $a_i$ depends really from $n$ variables (see [6]).

## V. Time Evaluation

We have implemented computer application, which uses family of graphs $RDE(n,K)$ for *private key* cryptography. To achieve high speed property, commutative ring $K = Z_{2^k}$, $k \in \{8, 16, 32\}$, with operations $+, \times$ modulo $2^k$. Parameter $n$ stands for the length of plaintext (input data) and the length of ciphertext. We mark by $G1$ the algorithm with $k = 8$, by $G2$ the algorithm with $k = 16$, and by $G4$ the algorithm with $k = 32$. So $Gi, i \in 1,2,4$ denotes the number of bytes used in the alphabet (and the size of 1 character in the string).

The alphabet for password is the same $K$ as for the plaintext. For encryption we use the scheme presented in section (4). The colour of vertex is its first coordinate.

If $u$ is the vertex, $p(u)$ is the colour of this vertex, and $\alpha$ is the character of password, then next vertex in the encryption path $v$ have the colour $p(v) = p(u) + \alpha$. All the next coordinates of $v$ are computed from (3) set of equations.

All the test were run on computer with parameters:

- AMD Athlon 1.46 GHz processor
- 1 GB RAM memory
- Windows XP operating system.

The program was written in Java language. Well known algorithms RC4 and DES which were used for comparison have been taken from Java standard library for cryptography purposes—*javax.crypto*.

### A. Comparison our algorithm with RC4

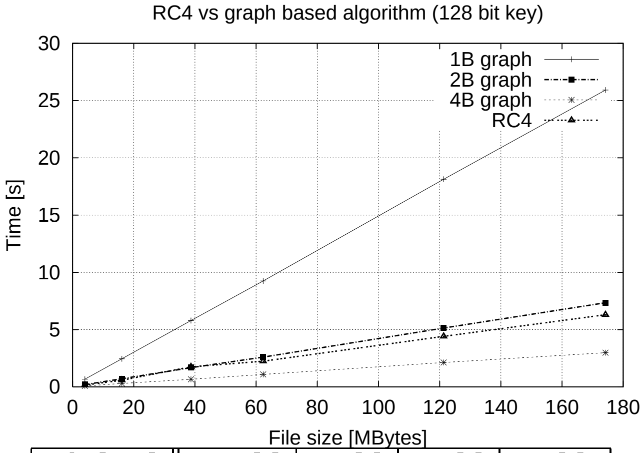
RC4 is a well known and widely used stream cipher algorithm. Protocols SSL (to protect Internet traffic) and WEP (to secure wireless networks) uses it as an option. Nowadays RC4 is not secure enough and not recommended for use in new system. Anyway we chose it for comparison, because of its popularity and high speed.

RC4 is not dependent on password length in terms of complexity, and our algorithm is. Longer password makes us do more steps between vertices of graph. So for fair comparison we have used fixed password length equal suggested upper bound for RC4 (16 Bytes).

TABLE I
Time grow for $\mathbf{A}_n E_{\bar{a}} \mathbf{A}_n^{-1}$ for chosen operator $\mathbf{A}_n$

| File [MB] | G1 [s] | G2 [s] | G4 [s] |
|---|---|---|---|
| | | | |
| 4 | 0.04 | 0.02 | 0.01 |
| 16.1 | 0.12 | 0.10 | 0.08 |
| 38.7 | 0.32 | 0.24 | 0.20 |
| 62.3 | 0.50 | 0.40 | 0.30 |
| 121.3 | 0.96 | 0.76 | 0.60 |
| 174.2 | 1.39 | 0.96 | 0.74 |

The mixing properties and speed comparison with DES the reader can find in [4]. The public key algorithms associated with the above dynamical system have been introduced in [11], [13].



| File [MB] | RC4 [s] | G1 [s] | G2 [s] | G4 [s] |
|---|---|---|---|---|
| 4 | 0.15 | 0.67 | 0.19 | 0.08 |
| 16.1 | 0.58 | 2.45 | 0.71 | 0.30 |
| 38.7 | 1.75 | 5.79 | 1.68 | 0.66 |
| 62.3 | 2.24 | 9.25 | 2.60 | 1.09 |
| 121.3 | 4.41 | 18.13 | 5.14 | 2.13 |
| 174.2 | 6.30 | 25.92 | 7.35 | 2.98 |

Fig. 1. RC4 vs high girth graph based algorithm (128 bit password)

## REFERENCES

[1] N. Biggs, *Algebraic Graph Theory* (2nd ed), Cambridge, University Press, 1993.

[2] N. Koblitz, *A Course in Number Theory and Cryptography*, Second Edition, Springer, 1994, 237 p.

[3] N. Koblitz, *Algebraic aspects of Cryptography*, in Algorithms and Computations in Mathematics, v. 3, Springer, 1998.

[4] J. Kotorowicz, V. A. Ustimenko, “ On the implementation of cryptoalgorithms based on agebraic graphs over some commutative rings”, *Condenced Matters Physics*, Proceedings of the international conferences “Infinite particle systems, Complex systems theory and its application,” Kazimerz Dolny, Poland, 2005-2006, 2008, vol. 11, N2(54), 347–360.

[5] F. Lazebnik F. and V. Ustimenko, “ Explicit construction of graphs with an arbitrary large girth and of large size”, *Discrete Appl. Math.* , 60, (1995), 275–284.

[6] F. Lazebnik, V. A. Ustimenko and A. J. Woldar, “A New Series of Dense Graphs of High Girth”, *Bull (New Series) of AMS*, v.32, N1, (1995), 73–79.

[7] E. H. Moore,“Tactical Memoranda“, *Amer. J. Math.*, v. 18, 1886, 264-303.

[8] R. Ore, *Graph Theory*, London, 1974.

[9] T. Shaska, W. C. Huffman, D. Joener and V. Ustimenko (editors),*Advances in Coding Theory and Cryptography*, Series on Coding Theory and Cryptology, vol. 3, 181-200 (2007).

[10] J. Seberry, J. Pieprzyk, *Cryptography: An Introducion to Computer Security*, Prentice Hall 1989, 379 p.

[11] V. Ustimenko, “ Maximality of affine group and hidden graph cryptsystems“, *Journal of Algebra and Discrete Mathematics*, October, 2004, v. 10, pp. 51–65.

[12] V. Ustimenko, *On the extremal graph theory for directed graphs and its cryptographical applications*, In: T. Shaska, W. C. Huffman, D. Joener and V. Ustimenko, Advances in Coding Theory and Cryptography, Series on Coding Theory and Cryptology, vol. 3, 181-200 (2007), 131–156.

[13] V. A. Ustimenko, “On the graph based cryptography and symbolic computations”, *Serdica Journal of Computing*, Proceedings of International Conference on Application of Computer Algebra, ACA-2006, Varna, N1 (2007), 131-186.

[14] V. A. Ustimenko, “ Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography”, *Journal of Mathematical Sciences*, Springer, vol. 140, N3 (2007), pp 412–434.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 303–310

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Empirically Tuning LAPACK's Blocking Factor for Increased Performance

R. Clint Whaley
Department of Computer Science
University of Texas at San Antonio
San Antonio, TX 78249
Email: whaley@cs.utsa.edu

***Abstract*—LAPACK (Linear Algebra PACKage) is a statically cache-blocked library, where the blocking factor (NB) is determined by the service routine ILAENV. Users are encouraged to tune NB to maximize performance on their platform/BLAS (the BLAS are LAPACK's computational engine), but in practice very few users do so (both because it is hard, and because its importance is not widely understood). In this paper we (1) Discuss our empirical tuning framework for discovering good NB settings, (2) quantify the performance boost that tuning NB can achieve on several LAPACK routines across multiple architectures and BLAS implementations, (3) compare the best performance of LAPACK's statically blocked routines against state of the art recursively blocked routines, and vendor-optimized LAPACK implementations, to see how much performance loss is mandated by LAPACK's present static blocking strategy, and finally (4) use results to determine how best to block nonsquare matrices once good square blocking factors are discovered.**

## I. Introduction

LAPACK [1] (Linear Algebra PACKage) is one of the most widely-used computational libraries in the world. LAPACK is the successor to the highly successful LINPACK[2] and EISPACK[3] packages. The main difference between LAPACK and its ancestor packages is that it blocks for the cache, which allows it to run orders of magnitude faster on modern architectures. Since linear algebra is computationally intensive, it is important that these operations, which are inherently very optimizable, run as near to machine peak as possible. In order to allow for very high performance with a minimum of LAPACK-level tuning, LAPACK does the majority of its computation by calling a lower-level library, the BLAS (Basic Linear Algebra Subprograms). The BLAS are in turn split into three "levels" based on how much cache reuse they enjoy, and thus how computational efficient they are. In order of efficiency, they are: Level 3 BLAS[4], which involve matrix-matrix operations that can run near machine peak, Level 2 BLAS[5], [6] which involve matrix-vector operations and are considerably slower, and Level 1 BLAS[7], [8], which involve vector-vector operations, and usually run at the speed of memory.

In order to maximize performance, LAPACK is therefore formulated to call the Level 3 BLAS whenever possible. LAPACK's main strategy for accomplishing this can be understood by examining its factorizations routines, QR, LU, and Cholesky. For each of these factorizations, LAPACK possesses two implementations: the first is an unblocked implementation that calls the Level 1 and 2 BLAS exclusively, which results in low performance. These routines are called DGEQR2/DGETF2/DPOTF2, respectively for QR/LU/Cholesky (assuming double precision). To get better performance, LAPACK has statically blocked versions of these routines (DGEQRF/DGETRF/DPOTRF), which calls the unblocked code on a subsection of the matrix, and then updates the rest of the matrix using the Level 3 BLAS (especially matrix multiply, DGEMM). The idea is that the updating of the rest of the matrix will run near peak when using a well-tuned BLAS implementation, and that this cost will dominate the unblocked factorization cost enough to allow LAPACK to be extremely computational efficient on any cache-based architecture.

Because LAPACK is such a large and complex library, allowing almost all the system tuning to occur in a smaller kernel library like the BLAS is critical, since LAPACK is too large for hand-tuners to make the individual routines computationally efficient. Through careful design, almost all platform-specific tuning has been moved to the BLAS, leaving LAPACK to concentrate on algorithm development. However, LAPACK possesses one key parameter that must be tuned to the platform to achieve maximal performance, which is the blocking factor (NB) used in LAPACK's statically blocked routines. To understand this further, we quickly overview the algorithm of DGETRF, which operates on an $M \times N$ matrix $A$: DGETF2 is called to factor an $M \times$ NB column panel of $A$ into $L$ and $U$, then an NB $\times (N - NB)$ row panel is updated using the Level 3 BLAS routine DTRSM, and these two results are then used to update the entire $(M - NB) \times (N - NB)$ trailing submatrix using DGEMM. This process is repeated, with the $M$ and $N$ above shrinking by NB each time, until the entire matrix has been factored. This type of blocked operation is roughly what occurs in a great many of LAPACK routines, including QR and Cholesky, though the details of the exact partitionings and routines used in updates vary widely.

NB cannot be chosen arbitrarily if good performance is to be maintained. NB=1 results in unblocked operation, and so defeats the whole purpose of blocked algorithms. Increasing NB typically allows the Level 3 updates to become more efficient,

This work was supported in part by National Science Foundation CRI grant SNS-0551504

as these operations have $O(N^3)$ computations which dominate the $O(N^2)$ memory references as $N$ is increased. However, we cannot set $\mathtt{NB} = N$, since this would wind up calling the unblocked code for the whole algorithm (no Level-3 updates left to do). Choosing a good NB is therefore a balancing act: making it large typically improves Level 3 performance, but as it grows, the unblocked code's performance becomes an increasing percentage of computational time, resulting in reduced performance. Therefore, the best NB is essentially unknowable a priori: it depends on the LAPACK routine being tuned, on the architecture being used, the relative performance of the Level 1 and 2 BLAS vs. Level 3 for the particular BLAS library being used, etc. The designers of LAPACK were aware of this, and so they did not hardwire the NB into the algorithm. Rather, most blocked LAPACK routines query a service routine called ILAENV for what blocking factor should be used. The LAPACK documentation stresses that users should tune this blocking factor in ILAENV for their problems of interest during install. Unfortunately, almost no users do so in practice: most are unaware that they should, and the remainder find it too hard to do. Therefore, the majority of LAPACK installations use the default blocking factors, which we show can yield significantly lower performance.

We have developed a framework for auto-tuning NB for LAPACK in order to fix this problem. This framework is part of our empirical tuning package, ATLAS [9], [10], [11]. ATLAS (Automatically Tuned Linear Algebra Package) provides empirically-tuned BLAS, and a few improved LAPACK routines. This new LAPACK tuning mechanism is part of ATLAS's empirical tuning framework, but it can be used to tune LAPACK for any BLAS, as this paper will describe.

The remainder of this paper is organized in the following way: Section II provides a quick overview of the empirical NB-tuning framework and service routines we have developed; Section III then presents results from empirically tuning LAPACK's factorization routines. We show that this can make a large difference in performance for a variety of platforms and BLAS implementations, and that these results point to a simple algorithmic change that could improve performance still further. Finally, Section IV summarizes our findings, and discusses future work.

## II. ATLAS's Empirical Tuning Framework for LAPACK

In this section we briefly overview the tools we developed for this investigation, as other BLAS users and researchers may find them useful. There are two main routines of interest, both of which appear in ATLAS's ATLAS/bin directory (assuming ATLAS version 3.9.2 or later).

The main tool is an LAPACK timer that can empirically tune NB for a variety of LAPACK routines, ATLAS/bin/lanbtst.c. This general-purpose timer links in a specialized ILAENV that allows the timer to change the ILAENV return value as part of the timing process. It can time arbitrary problem sizes, using arbitrary blocking factors, and be built against any mixture of (system LAPACK, ATLAS LAPACK, netlib LAPACK) + (system BLAS, ATLAS BLAS, F77BLAS). We presently support timing the three LAPACK factorizations: GETRF (LU), POTRF (Cholesky) and GEQRF/GERQF/GEQLF/GELQF (QR); adding additional routines is straightforward. Therefore, this timer/tuner can be used in a variety of ways to investigate performance, but here we use it to tune the blocking factor as $N$ changes. When used in this way, it generates either C macros or FORTRAN77 functions that take in the problem size, and return the best NB found by the empirical tuning for a problem of comparable size. The user can control what sizes of problems to tune, what range of NBs to try, how many times to repeat each timing, how many FLOPS are necessary to get reliable timings, etc. This timer automatically flushes the caches for realistic out-of-cache performance results, as described in [12]

To build the various LAPACK/BLAS combinations, the user fills in ATLAS's Make.inc BLASlib (system BLAS), FLAPACKlib (Fortran77/netlib LAPACK), and SLAPACKlib (system LAPACK). For instance, to tune LAPACK for the GotoBLAS installed in /opt/lib, you would fill in:

```
BLASlib = /opt/lib/libgoto_opteronp-r1.26.a \
          $(CBLASlib) $(ATLASlib)
FLAPACKlib = /opt/lib/libreflapack.a
SLAPACKlib = /opt/lib/libgoto_opteronp-r1.26.a \
             $(FLAPACKlib)
```

The LAPACK tuner's base name is x<pre>lanbtst, where <pre> is replaced by the precision prefix (eg., xdlanbtst for double precision). To this base name add a suffix indicating what variation should be timed. The suffix looks like: 'F_[f,s]l_[f,s,a]b. The first [f,s] before the 'l' chooses what LAPACK to use: fortran or system. The second part of the suffix determines what BLAS are linked in (thus the 'b' at the end): 's' links in the system BLAS, 'f' links in the Fortran BLAS (defined in the macro FBLASlib), and 'a' links in the ATLAS BLAS. To build a timer for ATLAS LAPACK with ATLAS BLAS, the make target is simply: 'xdlanbtst'. Thus, to build a tuner for the FORTRAN77 reference LAPACK linked to the GotoBLAS (assuming the above BLASlib setting), we would issue make xdlanbtstF_fl_sb. Calling the LAPACK tuner with invalid arguments or --help prints usage information.

The other tool of interest can be built with 'make xstattime'. This program reads in output from various ATLAS timers, and provides the results in a spreadsheet-friendly format. When timings are repeated multiple times, it also provides some basic statistical information (eg., average, max, min, standard deviation, etc.). If xstattime is used to compare two different runs, it uses a Student's T-Test [13] to compare whether the two means, from distributions with possibly different variances, are statistically different or not (this last feature requires a high number of samples to be reliable). Again, --help will provide more detailed usage information.

## III. Results

### A. Experimental Methodology

Version information: `gcc/gfortran 4.2`, ATLAS3.9.2 (labeled **atl**), ACML4.1.0 (**acm**), and GotoBLAS v2.91 (**got**). All performance results are gathered with a timer that flushes the caches, as described in [12]; therefore the small-case performance here may be much lower than those typically published, but more indicative of most real-world performance. Since MFLOP results are easier to understand in terms of peak performance, we report MFLOP rates rather than time. This could be problematic, in that some LAPACK routines (eg. GEQRF) do extra FLOPS as `NB` grows. In order to avoid this kind of problem, we compute the FLOP count for all algorithms as if they used the unblocked code, regardless of the blocking factor actually used. Therefore, these MFLOP results may be converted to time by simply by dividing by the FLOP count of the algorithm and inverting. The FLOP counts for (Cholesky, LU, QR) are very roughly $(\frac{N^3}{3}, \frac{2N^3}{3}, \frac{4N^3}{3})$, assuming square matrices. The exact FLOP counts can be found in LAPACK's `dopla.f`[14] routine.

Our main results are for two machines: a 2.4Ghz Core2Duo (abbreviated as **c2d**) and 2.2Ghz AMD Athlon-64 X2 (**ath**), both of which were running Kubuntu 8.04 (2.6 Linux kernel). On the **c2d**, we provide results for ATLAS[11] and Goto[15] BLAS: we are unable to provide results for Intel's MKL[16], since their libraries are not freely available for academic use. On the AMD system, we provide results for ATLAS, GotoBLAS, and AMD's ACML[17]. For both these machines, we time 64-bit libraries (which tend to have better performance than 32-bit). Unless otherwise specified, all timings were taken with a cycle-accurate wall-time, all results are the best of eight runs, with small problems being repeated within the timing interval to avoid resolution problems, as described in [12]. We also present a small subset of results for a 900Mhz Itanium 2 system. The Itanium is not featured more prominently because we only have ATLAS results on this machine: MKL was not available, and the GotoBLAS seg faulted during installation when compiled with `gcc 4.2` (we know GotoBLAS install without error with `gcc 3.x` on this platform, but were unable to install `gcc 3.x` on the Itanium in time for this paper). As it is, the limited Itanium results are provided mainly to demonstrate that this technique is in no way x86-specific. The Itanium timings use Linux's `gettimeofday` as the system timer.

### B. Tuning Speedups and Performance Overview

Figure 1 shows the performance in MFLOP for the appropriate BLAS's `DGEMM`, and netlib LAPACK's factorizations, using various BLAS. For each problem size in each figure, the first three bars are timings on the **ath**, while the last two are on **c2d**. The first column (white with black horizontal bars) uses ACML BLAS on **ath**, the second column (solid blue/gray) uses ATLAS BLAS on **ath**, the third column (slanting cross-hatching) uses GOTO BLAS on **ath**, the fourth column (textured solid blue/gray) uses ATLAS BLAS on **c2d**, and the final column (square cross-hatching) uses GotoBLAS on **c2d**.

The DGEMM timings of Figure 1(a) should provide an estimate on the peak performance the respectively libraries can achieve on these platforms. The Core2Duo has twice the theoretical peak of the Athlon-64, and so we see that the **c2d** is considerably faster regardless of the library. On the **ath**, the libraries' DGEMM performance is roughly equal, with ATLAS trailing ACML and GotoBLAS slightly (roughly 3%) for large problems. On the Core2Duo, the GotoBLAS have a noticable advantage, particularly for small problems.

Figures 1(b-d) show the performance of `DGEQRF` (QR), `DGETRF` (LU), and `DPOTRF` (Cholesky), respectively. Each of these timings is for the netlib LAPACK's factorization routine using the default blocking factor returned by the stock `ILAENV` (`NB` = 32, 64, and 64, respectively). It is the performance of these routines that we show can be improved with an empirical tuning of `NB`. On the **ath**, all libraries have similar performance, with the GotoBLAS doing generally better for small problems. We will see that tuning for the blocking strongly improves small-case performance, particularly for `ATLAS` and `ACML`, which are weak for small problems when using the default `NB`.

We then used the ATLAS infrastructure to tune the blocking factor for each operation on each architecture. We don't have room for full results, but blocking factors for all `DGEQRF` and some `DPOTRF` settings are shown in Table I. We give full results for `DGEQRF`, because neither the ATLAS nor GotoBLAS libraries supply an optimized version of this routine (they *do* supply optimized `DGETRF` and `DPOTRF`, as we will see in Section III-D). Therefore, the LAPACK version we tune here is actually the best one available when using these libraries. The `DGETRF` results are very similar to those for QR, but `DPOTRF` is very different, so we show full `DPOTRF` results for the Athlon-64 architecture.

For QR, the general trend is that the larger the problem, the larger the best blocking factor, up until a maximum is reached. This maximum represents the point at which the DGEMM performance gain from further increasing `NB` is offset by the performance loss due to spending greater time in the unblocked code. If we can optimize the unblocked part of the algorithm (a promising approach for doing this is outlined in Section III-C), then we can almost certainly increase performance further by using larger blocking factors for large problems.

`DPOTRF` looks very different, in that `NB` has apparently stopped growing only for ACML. One advantage Cholesky has over LU is that it doesn't require pivoting, which makes the unblocked portion of the algorithm much cheaper. In LU or QR, vastly increasing `NB` will speed up DGEMM, but the corresponding slowdown in the unblocked code will make it a net loss. Since the unblocked cost is relatively trivial in Cholesky, the block factor is free to grow much larger before the unblocked costs become prohibitive. There is further pressure in Cholesky for larger blocking factors: Cholesky's performance is strongly affected not only by DGEMM, but

TABLE I
OPTIMIZED NBS FOUND BY ATLAS'S EMPIRICAL TUNING STEP

| | DPOTRF | | | DGEQRF | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | ath64 | | | ath64 | | | c2d | | Itan2 |
| N | acm | atl | got | acm | atl | got | atl | got | atl |
| 25 | 12 | 8 | 12 | 12 | 4 | 4 | 4 | 4 | 4 |
| 50 | 24 | 8 | 24 | 12 | 8 | 8 | 12 | 12 | 8 |
| 75 | 28 | 8 | 28 | 12 | 8 | 12 | 12 | 16 | 8 |
| 100 | 28 | 4 | 36 | 12 | 4 | 16 | 8 | 12 | 8 |
| 125 | 28 | 8 | 32 | 12 | 12 | 16 | 12 | 16 | 8 |
| 150 | 52 | 16 | 52 | 12 | 12 | 16 | 12 | 16 | 8 |
| 175 | 28 | 16 | 60 | 16 | 12 | 24 | 12 | 16 | 8 |
| 200 | 28 | 16 | 68 | 12 | 12 | 24 | 16 | 16 | 16 |
| 250 | 28 | 16 | 84 | 16 | 12 | 28 | 16 | 24 | 16 |
| 300 | 16 | 16 | 76 | 24 | 28 | 28 | 12 | 24 | 16 |
| 350 | 16 | 52 | 88 | 28 | 28 | 32 | 56 | 28 | 24 |
| 400 | 16 | 24 | 144 | 28 | 28 | 32 | 56 | 28 | 40 |
| 450 | 52 | 52 | 96 | 24 | 28 | 32 | 56 | 32 | 24 |
| 500 | 56 | 24 | 88 | 32 | 28 | 36 | 56 | 40 | 24 |
| 600 | 80 | 52 | 88 | 28 | 28 | 40 | 56 | 40 | 24 |
| 700 | 112 | 52 | 144 | 32 | 36 | 44 | 56 | 40 | 24 |
| 800 | 128 | 52 | 144 | 32 | 52 | 52 | 56 | 48 | 24 |
| 900 | 192 | 112 | 192 | 32 | 52 | 52 | 56 | 44 | 40 |
| 1000 | 128 | 112 | 192 | 56 | 52 | 52 | 56 | 52 | 40 |
| 1200 | 192 | 128 | 192 | 56 | 52 | 52 | 56 | 64 | 40 |
| 1400 | 192 | 160 | 192 | 56 | 52 | 52 | 56 | 72 | 56 |
| 1600 | 192 | 160 | 192 | 56 | 52 | 52 | 56 | 80 | 56 |
| 1800 | 192 | 160 | 192 | 56 | 52 | 52 | 56 | 80 | 56 |
| 2000 | 192 | 224 | 224 | 56 | 52 | 52 | 56 | 96 | 56 |

also DSYRK. DSYRK's performance, like DGEMM, grows strongly with problem size.

We note that the GotoBLAS DGEQRF's NB on the **c2d** is still growing at our last problem size, though it is nowhere near as large as the Cholesky block. There are probably at least two factors in play here. The first is that such an excellent DGEMM (Goto achieves as much as 94% of theoretical peak) can hide quite a bit of poor unblocked performance before being washed out. The second reason the GotoBLAS's blocking factor is still growing is linked to the fact that the GotoBLAS block for the Level 2 cache, which we discuss further below.

Figures 2(b-d) show the % improvement from empirically tuning NB for a range of problem sizes for each BLAS library; each of Figures 2(b-d) shows the % improvement over the MFLOP rates provided in Figures 1 (b-d), respectively. More specifically, the X-axis of Figure 2 is computed as: $100.0 \times \left(\frac{\texttt{NB-tuned factorization performance}}{\texttt{default-blocked factorization performance}}\right)$. Since LAPACK's default NB is chosen to optimize mid-range problems, it is unsurprising that the general trend is that tuning NB helps most for small and large problems, with less benefit generally in the middle areas. We notice that the tuning is particularly helpful in the small-case problems for ACML and ATLAS, which displayed relatively poor performance in this area when using the default blocking factors. For large problems, we see that the **goto** BLAS on the Core2Duo almost always show substantial speedups. This is probably due to the fact that the GotoBLAS block for the L2 cache [18], and thus utilizing a larger block factor as $N$ is increased allows the factorization's GEMM calls to perform nearer peak. In Figure 2 (b), we see that ATLAS on the Core2Duo shows substantial speedups across the entire range. This is mainly due to their being a mismatch between the default NB of 32, which is not large enough for ATLAS's DGEMM to reach its full potential. All in all, we observe substantial speedups for both small (as high as 30%) and large (as high as 20%) problems.

In order to demonstrate that there is nothing x86-specific in this tuning advantage, Figure 2 (a) shows the improvement from tuning LAPACK's factorizations using the ATLAS BLAS on a 900Mhz Itanium 2. We see a huge (as much as 75%!) improvement for small cases when NB is tuned; large problems are improved substantially (as much as 15%).

With these results, we see that the empirical tuning of LAPACK's factorization routines provides substantial speedups, and it seems this should hold for most of LAPACK's blocked routines. Therefore, we strongly believe that this is a simple, general technique that can be used to markedly increase the performance of an overwhelming majority of LAPACK routines.

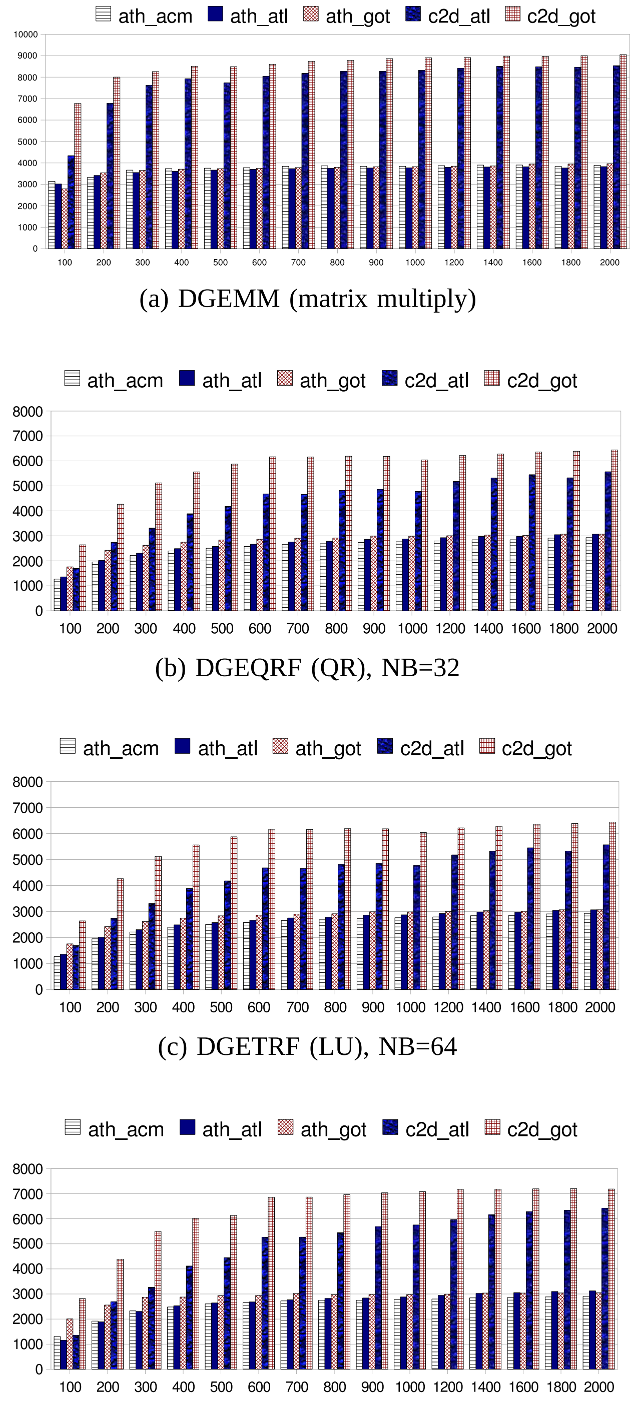
### C. *Tuning NB for Non-Square Problems*

Many LAPACK routines operate on non-square matrices, and some operate almost exclusively on highly rectangular arrays. In particular, many service routines operate on matrices where one dimension is constrained to the blocking factor of a higher-level routine (for instance, N is always 64), but the other dimension is unconstrained (and so might be something like 1000). For this discussion, the constrained (small) matrix dimension will be referred to as the *minor dimension*, while the unconstrained dimension is the *major dimension*. Both LU (DGETRF) and QR (DGEQRF) can operate on nonsquare matrices, so we use these operations to probe nonsquare behavior.

One obvious idea is to attempt to use the blocking factors discovered in the square matrix tuning for non-square. Obviously, we can't use the major dimension to select the tuned NB, since blockings greater than the smallest dimension of the matrix don't make sense (assuming the square blockings used in LAPACK). Therefore, our first question is: how much performance is lost if we use our square-tuned NBs, where our decision is based on the minor dimension? Table II provides the answer to this question.

In Table II, we have surveyed all three libraries on the **ath** architecture, with varying operations. The first column gives the architecture, library, and LAPACK operation, the second column indicates which dimension (matrix is $M \times N$) is constrained, and to what value it is constrained to. The third column indicates the blocking suggested by our square-tuned NBs, where NB choice is based on the indicated minor dimension. Then, for this minor dimension, we tune the blocking factor for a variety of major dimension sizes. For each size, we report the best blocking factor found, and in parenthesis, the percentage of its performance that the square-tuned NB achieves.

While the details change with architecture, operation, and matrix dimensions, several important trends are immediately obvious. The first is that if the operation is called on these



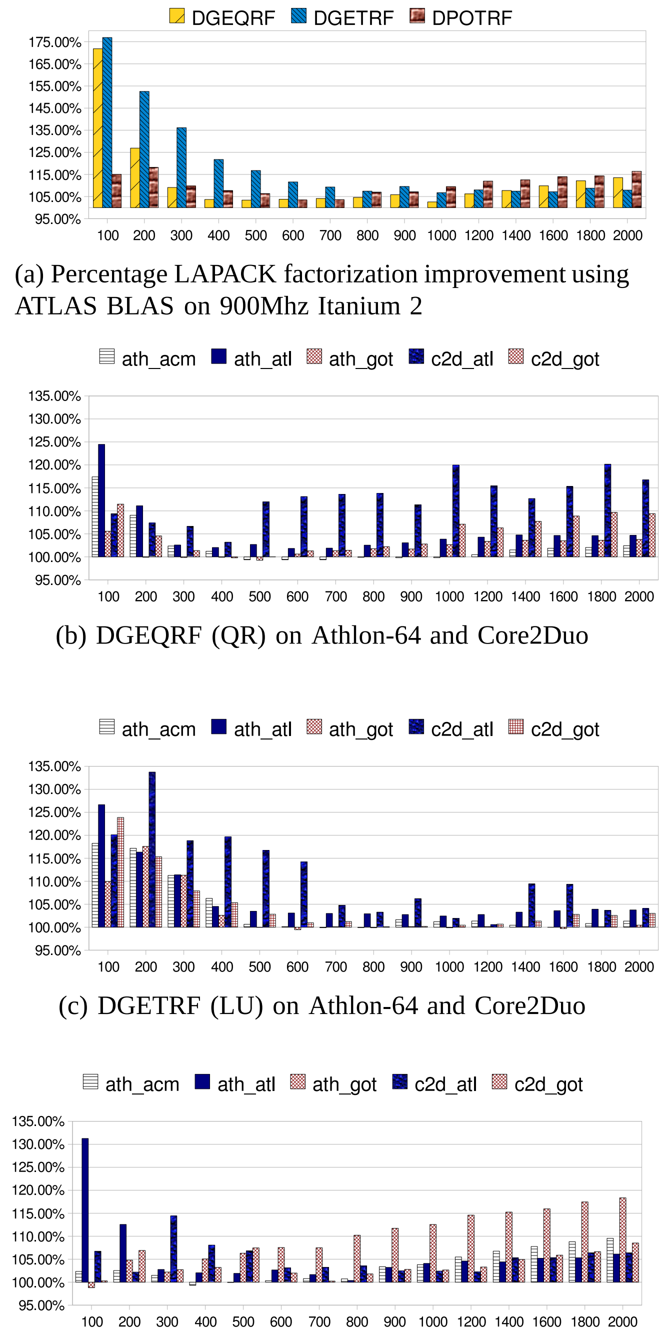
(a) DGEMM (matrix multiply)

(b) DGEQRF (QR), NB=32

(c) DGETRF (LU), NB=64

(d) DPOTRF (Cholesky), NB=64

Fig. 1. Basic DGEMM and LAPACK Factorization Performance in MFLOPS for ACML, ATLAS, and GOTO on 2.2 Ghz Athlon (first three bars) and ATLAS and GOTO on 2.4Ghz Core2Duo (last two bars). Factorizations use LAPACK's default blocking.



(a) Percentage LAPACK factorization improvement using ATLAS BLAS on 900Mhz Itanium 2

(b) DGEQRF (QR) on Athlon-64 and Core2Duo

(c) DGETRF (LU) on Athlon-64 and Core2Duo

(d) DPOTRF (Cholesky) on Athlon-64 and Core2Duo

Fig. 2. NB-optimized LAPACK factorization performance as a percentage of default NB performance.

TABLE II
SELECTED RESULTS FOR TUNING NB FOR NON-SQUARE PROBLEMS. FOR EACH PROBLEM SIZE, WE REPORT BOTH THE BEST NB, AND, IN PARENTHESIS, THE % OF ITS PERFORMANCE THAT WE GET USING THE SQUARE-TUNED NB WITH N = MIN(M,N).

| **arch/** | **minor** | **square** | **Major Dimension** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **lib/op** | **dim** | **NB** | 50 | 100 | 300 | 500 | 1000 | 2000 | 10000 |
| ath/acml/QR | M=32 | 12 | 1(74%) | 1(84%) | 8(100%) | 12(100%) | 12(100%) | 12(94%) | 16(93%) |
| ath/acml/QR | N=32 | 12 | 1(76%) | 1(90%) | 8(100%) | 8(97%) | 8(92%) | 8(92%) | 8(83%) |
| ath/acml/QR | N=64 | 12 | 1(82%) | 16(98%) | 8(x4%) | 8(99%) | 8(97%) | 8(100%) | 8(88%) |
| ath/goto/QR | M=32 | 4 | 1(79%) | 12(89%) | 16(81%) | 16(78%) | 24(64%) | 24(61%) | 24(65%) |
| ath/goto/QR | N=32 | 4 | 1(73%) | 1(78%) | 1(84%) | 16(85%) | 8(89%) | 8(92%) | 8(92%) |
| ath/atlas/LU | N=32 | 8 | 8(100%) | 8(100%) | 8(100%) | 12(99%) | 12(97%) | 12(95%) | 8(100%) |
| ath/atlas/LU | N=64 | 16 | 8(98%) | 16(100%) | 12(94%) | 12(92%) | 12(92%) | 12(94%) | 12(84%) |

highly nonsquare shapes, then it makes sense to tune for it explicitly, as the best rectangular NB may substantially outperform the square-tuned cases (this is particularly true for the GotoBLAS). Even for a given minor dimension, we see that the best NB often grows as the major dimension is increased; this observation alone is a strong indication that even very well-tuned square-case NBs will not provide optimal non-square results as the gap between minor and major dimensions grow. Another take-home lesson from this table is that the best tuning parameter for a column panel (eg, minor dimension is N) is not necessarily the same as that for an equal-sized row panel (minor dimension is M): row panels tend to need larger NBs, (this would provide more TLB reuse, which is not a problem for column panels).

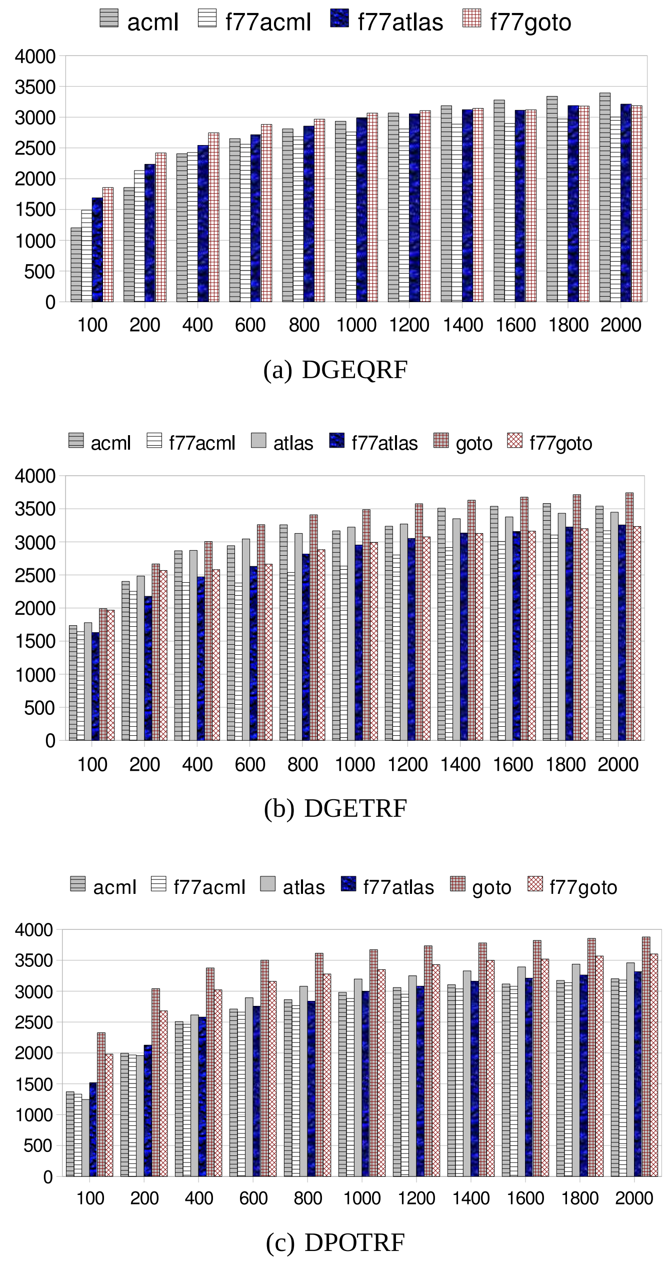
This table also points us to an obvious way to speed up LAPACK beyond merely tuning the blocking factor. Note that choosing NB=1 in these timings essentially means that the unblocked service routines (DGETF2 for DGETRF and DGEQR2 for DGEQRF), which call the Level 1 and 2 BLAS only, are used instead of the Level-3 BLAS-based blocked implementations. Blocked LAPACK routines typically call their unblocked equivalents with one dimension (usually $M$) set to the size of the original problem (in the first step; its size will decrease by NB during each iteration), while the second dimension is set to NB. DGETRF will factor an $M \times \mathtt{NB}$ portion of $A$ into L and U using the unblocked routine DGETF2 in its first step. Now, if we assume NB is 32 or 64, then Table II actually shows the performance we can expect from a NB-tuned DGETRF (DGEQRF) routine when used in place of DGETF2 (DGEQR2). Anytime the best NB $\neq 1$, then it would be faster to have the blocked routine call a second blocked routine, which uses the smaller blocking factor indicated in the appropriate column of the table. Unfortunately, we don't have room to indicate how much of a performance win this would be for all table entries, but we can bring out a few highlights: For $N = 1000, N = 64$, the tuned blocked implementation is (1.67,1.34,1.76) times faster than the unblocked implementation of (**ath**/ACML/QR, **ath**/Goto/QR, **ath**/ATLAS/LU), respectively. Note that this would allow LAPACK's statically blocked routines to block the "unblocked" code, just as the recursive formulation does, which is key, in that it allows the outer implementation to use larger blocking factors, and thus achieve a higher percentage of DGEMM peak (i.e., the speedup from this technique can go beyond that of merely speeding up the "unblocked" code, since faster "unblocked" code means that we can afford a large NB, which may substantially improve DGEMM performance).

### *D. Tuned Reference vs. Library-Provided LAPACK*

The surveyed optimized libraries (ACML/ATLAS/Goto) provide LAPACK routines in addition to the BLAS. The natural question is how well do the empirically tuned reference LAPACK routines compare with their optimized-library counterparts. The answer is surprisingly well, as we can see in Figure 3. In this figure, we compare the performance of various implementations of DGEQRF, DGETRF, and DPOTRF on the Athlon-64. The native implementations (i.e. LAPACK routines present in the optimized library) are annotated with only the library name (eg., "acml"), while the version that use the netlib FORTRAN77 reference implementations with an empirical tuning step have f77 prefixed to the library that provided the BLAS (eg., "f77acml" is the reference implementation of the LAPACK routine from netlib, with the blocking factor empirically tuned for the ACML BLAS).

Figure 3 (a) shows the performance of various implementations of DGEQRF on selected problem sizes on the Athlon-64 (the number of problem sizes shown has been reduced as an aid to legibility). As mentioned, neither GotoBLAS nor ATLAS provides a native QR, so for these libraries we show only the performance of the tuned netlib LAPACK. ACML provides a native QR, and so we see its performance as the first bar for each problem size. For small problems, it is the worst performing QR, with the best performance coming from the empirically tuned LAPACK using the GotoBLAS. For large problems, ACML's native implementation provides the best performance.

For LU factorization (Figure 3 (b)), all libraries provide a native routine, and the native routines beat their tuned netlib versions across the entire surveyed range. For small problems, there is little gap between the f77 and native implementations, but this gap gets noticeably larger for large problems, particularly for the GotoBLAS library. The same basic trend is seen for Cholesky (Figure 3 (c)), with the marked exception of the GotoBLAS-optimized DPOTRF. The GotoBLAS provides a noticeably superior Cholesky regardless of LAPACK used,



(a) DGEQRF

(b) DGETRF

(c) DPOTRF

Fig. 3. Tuned reference vs. library-provided (native) LAPACK on the Athlon-64. Native routines have same pattern as reference, but the background color is light gray rather than white (or blue/dark gray for ATLAS). From left to right: ACML-LA+ACML-BLAS, NETLIB-LA+ACML-BLAS, ATLAS-LA+ATLAS-BLAS, NETLIB-LA+ATLAS-BLAS, GOTO-LA+GOTO-BLAS, NETLIB-LA+GOTO-BLAS (there is no ATLAS- or GOTO-provided DGEQRF, and so these bars do not show up in figure (a)).

but its native implementation is commandingly better for small problems.

These results are fairly easy to understand for ATLAS: ATLAS uses recursive implementations (based on the work of [19], [20], [21]) of LU and Cholesky, and for large problems it is expected that the much larger DGEMM calls seen at the top of the recursion will naturally lead to the performance advantage we see for large problems. Recursion is essentially the only advantage ATLAS has over netlib LAPACK: we don't currently do any system tuning at this level. For Cholesky, the ATLAS `DPOTRF` actually loses to the empirically-tuned netlib LAPACK. This is probably due to recursive overhead: remember that in LU and QR, the recursion cuts only the $N$ dimension, while in Cholesky it cuts both dimensions, which leaves very few operations over which to amortize the cost of the recursive calls. It seems likely that if the unblocked code in LAPACK was replaced with blocked code, as discussed in Section III-C, that the netlib LAPACK could win for small problems on LU as well, again mostly due to overhead savings. It seems unlikely that even this improved version could compete for very large problems: the recursive advantage should grow with problem size, so eventually recursion will begin to win for large enough matrices. However, the gap is not too large for the problem sizes surveyed here, so with better unblocked code, it is possible the netlib LAPACK could be made to run competitively across this entire range.

It is difficult to know what the other libraries are doing to produce these performance curves. GotoBLAS does get better `DSYRK` (particularly small-case) performance than ATLAS, which in turn gets much better `DSYRK` performance than ACML, which probably helps explain `DPOTRF` results. GotoBLAS's small-case performance advantage is large enough that there are probably additional factors needed to explain this gap, however.

## IV. Summary and Future work

We have presented a survey of factorization performance on two important architecture families (with a few results from a third). Using our empirical tuning system, we have demonstrated that tuning `NB` can provide substantial performance improvements regardless of BLAS used or the architecture it runs on. This simple tuning can be extended to speed up an overwhelming majority of LAPACK routines, in addition to the factorizations discussed here. We note that many routines may be further improved by rewriting them as recursive algorithms, but this is not currently possible for several important LAPACK routines. Therefore, empirically-tuned `NB` LAPACK implementations of such routines would actually represent the best available implementations today. Even in the case where recursive formulations are possible, we have shown that statically-blocked LAPACK performance can be made competitive with these techniques. We have also investigated tuning for nonsquare matrices, which could improve the performance of many of LAPACK's service routines, and this investigation has pointed out a straightforward adaptation of LAPACK's present strategy that can yield further performance improvements without requiring LAPACK to be rewritten in languages supporting recursion.

### A. Future work

In this paper, we concentrated on the factorizations, since they are probably the most widely used routines in the library (they are called by many higher-level routines, in addition to being called directly), in addition to being well understood routines which possess optimized implementations we can compare against. The tuning framework can be easily extended to other operations, and we plan to do so (we are particularly interested in tuning additional routines used in finding eigenvalue/vectors). The framework already allows other data types/precisions to be tuned, and probing the effect of varying type and precision on optimal blocking is an experiment worth doing. This investigation concentrated on serial performance: an obvious extension of great interest would be to see how results vary when the same libraries are used, but threading is enabled. Finally, our investigations pointed to significant speedups if we block the present unblocked codes with the small blocking factors discovered in our rectangular tuning. This should be investigated for the factorizations, and see how the resulting performance stacks up against state-of-the-art recursive formulations. If successful here, we should see if this technique can be used on some of the LAPACK routines that do not currently possess recursive formulations.

## REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. Philadelphia, PA: SIAM, 1999.

[2] J. Dongarra, J. Bunch, C. Molar, and G. Stewart, *LINPACK Users' Guide*. Philadelphia, PA: SIAM, 1979.

[3] B. Smith, J. Boyle, J. Dongarra, B. Garbow, Y. Ikebe, V. Klema, and C. Molar, "Matrix Eigensystem Routines - Eispack Guide, Second Edition," in *Matrix Eigensystem Routines - Eispack Guide*, ser. Lecture Notes in Computer Science, no. 6. Springer-Verlag, 1976.

[4] J. Dongarra, J. D. Croz, I. Duff, and S. Hammarling, "A Set of Level 3 Basic Linear Algebra Subprograms," *ACM Transactions on Mathematical Software*, vol. 16, no. 1, pp. 1–17, 1990.

[5] J. Dongarra, J. D. Croz, S. Hammarling, and R. Hanson, "Algorithm 656: An extended Set of Basic Linear Algebra Subprograms: Model Implementation and Test Programs," *ACM Transactions on Mathematical Software*, vol. 14, no. 1, pp. 18–32, 1988.

[6] ——, "An Extended Set of FORTRAN Basic Linear Algebra Subprograms," *ACM Transactions on Mathematical Software*, vol. 14, no. 1, pp. 1–17, 1988.

[7] R. Hanson, F. Krogh, and C. Lawson, "A Proposal for Standard Linear Algebra Subprograms," *ACM SIGNUM Newsl.*, vol. 8, no. 16, 1973.

[8] C. Lawson, R. Hanson, D. Kincaid, and F. Krogh, "Basic Linear Algebra Subprograms for Fortran Usage," *ACM Transactions on Mathematical Software*, vol. 5, no. 3, pp. 308–323, 1979.

[9] R. C. Whaley, A. Petitet, and J. J. Dongarra, "Automated empirical optimization of software and the ATLAS project," *Parallel Computing*, vol. 27, no. 1–2, pp. 3–35, 2001.

[10] R. C. Whaley and A. Petitet, "Minimizing development and maintenance costs in supporting persistently optimized BLAS," *Software: Practice and Experience*, vol. 35, no. 2, pp. 101–121, February 2005, `http://www.cs.utsa.edu/˜whaley/papers/spercw04.ps`.

[11] ——, "Atlas homepage," http://math-atlas.sourceforge.net/.

[12] R. C. Whaley and A. M. Castaldo, "Achieving accurate and context-sensitive timing for code optimization," Accepted for publication in *Software: Practice and Experience*, 2008.

[13] R. Walpole, R. Myers, S. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, 7th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[14] LAPACK, "`LAPACK/TIMING/LIN/dopla.f`," http://ib.cnea.gov.ar/~fiscom/Libreria/lapack_blas/source/TIMING/LIN/dopla.f.

[15] K. Goto, "Gotoblas homepage," http://www.tacc.utexas.edu/resources/software/.

[16] Intel, "Mkl homepage," http://www.intel.com/cd/software/products/asmo-na/eng/307757.htm.

[17] AMD, "Acml homepage," http://www.amd.com/acml.

[18] K. Goto and R. A. van de Geijn, "Anatomy of high-performance matrix multiplication," Accepted for publication in *Transactions on Mathematical Software*, 2008.

[19] S. Toledo, "Locality of reference in lu decomposition with partial pivoting," *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 4, 1997.

[20] F. Gustavson, "Recursion leads to automatic variable blocking for dense linear-algebra algorithms," *IBM Journal of Research and Development*, vol. 41, no. 6, pp. 737–755, 1997.

[21] F. Gustavson, A. Henriksson, I. Jonsson, B. Kågström, and P. Ling, "Recursive blocked data formats and blas's for dense linear algebra algorithms." in *Applied Parallel Computing, PARA'98*, ser. Lecture Notes in Computer Science, No. 1541, B. Kågström, J. Dongarra, E. Elmroth, and J. Waśniewski, Eds., 1998, pp. 195–206.

# Computational Linguistics—Applications

ACCORDING to the European Commission, Human Language Technologies are one of the key research areas for the upcoming years. The Computer Linguistics—Applications Workshop is in part a response to the fast-paced progress in the area.

The workshop will focus on the applied aspect of Computer Linguistics, i.e. the practical outcome of modeling human language.

The aim of the Applied Computer Linguistics is to enable the user to communicate with the computer in his/her native language. This may be useful for retrieving information from databases or raw texts as well as controlling a robot (virtual or real). Computer Linguistics may help communicate people with each other by means of computerized translation. Computer Linguistics is applied to make the full use of the Internet: we need software that can handle texts if we want to find the information we need in the web.

Furthermore, because of its location within the framework of the IMCSIT conference, we hope to initiate dialog between researchers involved in Computer Linguistics and other areas of information technology to build bridges between disciplines and find a way of applying CL-based tools in other areas. In this spirit, we invite papers that present research on all aspects of Natural Language Processing such as (this list is not exhaustive):

- machine translation and translation aids
- proofing tools
- semantic ontologies in computer linguistics
- lexical resources
- POS-tagging
- corpus-based language modeling
- extraction of linguistic knowledge from text corpora
- ambiguity resolution
- parsing issues
- information retrieval
- text classification

In addition to theoretical papers, we invite all participants to present practical demonstrations of existing tools. We require the papers to include a section describing an existing tool (or a prototype), which demonstrates the theory.

We intend the workshop to be a demo session. In the first part of the workshop the authors will shortly (no longer than 5 minutes each) demonstrate their tools to the audience. For the second part we will arrange environment in which authors will be able to present their software and discuss it with other conference participants.

The best demonstrations will be elected to be shown to the audience of the conference at a plenary session.

## International Programme Committee

**Maria Gavrilidou,** Institute for Language and Speech Processing, Marousi Greece

**Filip Graliński,** Poleng Ltd,Poland

**Ales Horak,** Masaryk University Brno, Czech Republic

**Anna Korhonen,** Natural Language and Information Processing (NLIP) Group University of Cambridge, UK

**Karel Pala,** Masaryk University, Czech Republic

**Gabor Proszeky,** Morphologic Hungary

**Jakub Piskorski,** EU, JRC, Web and Language Technology, Italy

**Zygmunt Saloni,** University of Warmia and Mazury, Poland

**Pavel Rychly,** Masaryk University, Czech Republic

**Kiril Simov,** Linguistic Modelling Laboratory, IPP, BAS, Bulgaria

**Stan Szpakowicz,** SITE, University of Ottawa

## Organizing Committee

Chair **Krzysztof Jassem,** UAM, Poland

**Adam Przepiórkowski,** ICS PAS, Poland

**Maciej Piasecki,** Wroclaw University of Technology, Poland

**Piotr Fuglewicz,** TIP, Poland

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 313–319

ISBN 978-83-60810-14-9
ISSN 1896-7094

# The impact of corpus quality and type on topic based text segmentation evaluation

Alexandre Labadié
LIRMM
161, rue Ada
34391 Montpellier Cedex 5
Email: labadie@lirmm.fr

Violaine Prince
LIRMM
161, rue Ada
34391 Montpellier Cedex 5
Email: prince@lirmm.fr

***Abstract*—In this paper, we try to fathom the real impact of corpus quality on methods performances and their evaluations. The considered task is topic-based text segmentation, and two highly different unsupervised algorithms are compared: $C99$, a word-based system, augmented with $LSA$, and $Transeg$, a sentence-based system. Two main characteristics of corpora have been investigated: Data quality (clean vs raw corpora), corpora manipulation (natural vs artificial data sets). The corpus size has also been subject to variation, and experiments related in this paper have shown that corpora characteristics highly impact recall and precision values for both algorithms.**

## I. Introduction

THERE are several distinct tasks labeled as ‘text segmentation’, among which **topic-based text segmentation** is the particular process that tries to find the topical structure [8] of a text and thus provide a possible thematic decomposition of a given document [17]. It relies on the evidence that most texts do not talk about only one topic and the longer the documents, the more topics they include. There are several definitions of a topic in the relevant literature. Generally speaking, a topic is: *the subject matter of a conversation or discussion*. In linguistics, it is defined as: *the part of the proposition that is being talked about (predicated)*. In this paper we do not consider the formal definition of topic, but the more commonly admitted definition that a topic of a text is *what talking is about*. So, the goal of an automated text segmentation could be simplified into dividing a text in segments, each sentence of which ’talks about’ the same subject. Segments are supposed to be distinct, and coherent [14].

Text segmentation, whether topic-based or not, has been thoroughly evaluated in several campaigns (in several TREC (Text Retrieval Evaluation Conference) editions, as well as in its French equivalent DEFT (Défi Fouille de Textes), especially in the DEFT 2006 session devoted to text segmentation [1]). However, if protocols (concatenating texts like in [7]), methods [2], and measure metrics [16] have been evoked, the corpora features have not been at the center of the attention. Apart from some general characteristics such as corpus lengths (in words, sentences or MBytes), or origin (e.g. news [20], newspapers or magazine articles [9]), corpora seem to have escaped their own evaluation procedure.

As participants in several text retrieval evaluation campaigns, we have been accustomed to dealing with corpora that were provided by the challenge organizing committees, and thus, not tailored for our demonstration needs. We have noticed that, the same algorithm, run on different corpora, had its results highly impacted by two items:

- The ’cleanliness’ of the corpus: Does it contain typos, unreadable words, several punctuation marks, unanticipated abbreviations, ill-formed sentences? All these elements could prevent some algorithms, especially those relying on NLP techniques such as syntactic and semantic analysis, from obtaining the results they are supposed to produce.
- The ’naturality’ of the corpus: Is it an artificially generated set of words, obtained through concatenation of segments or texts of different origins, or is it a real document or collection of documents, written in a given style, addressing a given subject, and so forth? Artificial corpora could favor techniques sensitive to clean cuts in topics, whereas natural corpora would introduce a higher difficulty, since transitions are smoother, and so topic shifting more difficult to detect.

In this paper, we try to fathom the real impact of corpus quality on the performances of topic-based segmentation methods. We have set up an experiment using two unsupervised segmenting algorithms, Choi’s $c99$ [7], a renowned segmenting method, and $Transeg$ [10], [18]. We chose unsupervised methods because they are not attuned to corpus idiosyncrasies as supervised methods are: Corpus adjustment by learning would have prevented us from studying the impact of its quality. The second reason for this choice is that both algorithms highly differ methodologically: The first is lexical, based on words frequencies, and neglects syntactical or rhetorical information, the second is more sentence or segment oriented, based on syntactic parsing and sentence semantics calculus, and is not sensitive to frequencies. The difference is crucial. A pending question is always the robustness issue: Would a lexical based system be less sensitive to an ’raw’ corpus than a sentence based system? Would it be more sensitive to an artificial corpus of concatenated texts, where differences in topics are neat and precise? This would mark a differential approach to quality sensitivity, and we wanted to know whether this feeling was justified or not.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

In section 2, we present both methods, their features and characteristics. In section 3, we describe our experiment and the working hypothesis that we have tried to evaluate. As it will be seen in results comments(section 4) and in conclusion (section 5), the impact of on precision/recall performances of both algorithms isn't to be ignored. But interesting differences are pointed out when matching both items (i.e. clean/dirty vs natural/artificial) distinct values. We hope this will help evaluation campaigns organizers shape up test corpora that will be as reliable and as discriminant as possible for the running methods they intend to evaluate.

## II. A brief overview of $c99$ and $Transeg$

In this section we present the two methods we compared during our experiments: the well known $c99$ algorithm [6] and $Transeg$ the method we are currently developing. Both are unsupervised and thus corpus free approaches.

### A. C99

Developed by Choi, $c99$ is a text segmentation algorithm strongly based on the lexical cohesion principle [15]. It is, at this time, one the best and most popular algorithms in the domain [2], which convinced us to choose it as a baseline for our experiments.

$C99$ uses a similarity matrix of the text sentences. First projected in a word vector space, sentences are then compared using the cosine similarity measure (by the way, the most used measure). Similarity values are used to build the similarity matrix. More recently, Choi improved $c99$ by using the Latent Semantic Analysis (LSA) achievements to reduce the size of the word vector space [7].

The author then builds a second matrix known as the *rank matrix*. The latter is computed by giving to each cell of the similarity matrix a rank equal to the number of cases around the examined one (in a layer) which have a lesser similarity score. This rank is normalized by the number of cases that were really inside the layer to avoid side effects.

$C99$ then finds topic boundaries by recursively seeking the optimum density of matrices along the rank matrix diagonal. The algorithm stops when the optimal boundaries returned are the end of the current matrix or, if the user gave this parameter to the algorithm, when the maximum number of text segments is reached. By definition, $c99$ always retrieves the first sentence of a text as a the beginning of a new topic (which is obviously true). Choi's original experiments in both cited papers use an 'artificial' corpus, created by concatenating multiple texts. So, retrieving text boundaries in a concatenation and segmenting topically a text have been considered as equivalent tasks by the authors.

All experiments in this paper have been conducted using the original latest version of the algorithm (it can be found at http://myweb.tiscali.co.uk/freddyyychoi/) to avoid any implementation bias. So testing $c99$ on natural, non concatenated texts provides information about its behavior in an environment different from the original protocol.

### B. Transeg

$Transeg$ is also based on a vectorial representation of the text and on a precise definition of what a *transition* between two text segments should be. It has been developed with a sentence parser, and until now, experiments and results have been obtained for the French language. A shifting to any other language is naturally possible, provided that a syntactic parsing occurs and the language words are dipped into a Roget-based representation. Since this system has not yet been as widely described as $c99$, we focus on its principles for the sake of information.

*1) Text Vector Representation:* The first step is to convert each text sentence into a semantic vector obtained using the French language parser SYGFRAN [3]. Vectors are mathematical representations of the Roget Thesaurus indexing each English word with a set of $1043$ concepts [19], but 'exported' to French, through the local Roget equivalent, the Larousse thesaurus [11] . Sentence vectors are recursively computed by linearly combining sentence constituents, in turn computed by linearly combining word vectors. The weights of each word vectors are computed according to a formula relying on a constituents and dependencies syntactic analysis (The formula is given in [4]). So, sentence vectors bear both the semantic and the syntactic information of the sentence, but are not sensitive to words frequencies.

*2) Transition zones and boundaries: How to detect topic shifts:* In well written structured texts, the transition between a topic and the next one is not abrupt. An author should conclude one topic before introducing another.This specific part of text between two segments is waht we call a **transition zone**. Ideally, the transition zone should be composed of two sentences:

- The last sentence of the previous segment.
- The first sentence of the beginning new segment.

$Transeg$ tries to identify these two sentences in order to track topic boundaries.

*a) Transition score and the beginning of a new segment:* The **transition score** of a sentence represents its likelihood of being *the first sentence of a segment*. Each sentence of the text is assumed to be the first of a $10$ sentences long segment. This 'potential segment' is then compared with another one composed by the $10$ preceding sentences. The $10$ sentences size was chosen by observing results on the training corpus of French political discourses in the DEFT06 competition, segmented by human experts. Competitors such as [10] noticed that the average size of a segment was around $10$ sentences $(10.16)$ with a $\sigma$ of $(3.26)$. So they decided to use this empirical value as the standard segment size. However, this value has no impact on boundaries detection. Any other might fit as well.

To compute the score of each sentence, $Transeg$ slides a $20$ sentences long window along the text, considering each half of the window as a potential segment. The latter is then represented by one vector, calculated as a weighted barycenter of its sentence vectors (which are designated as centroid in figure 1). Stylistic information was added by giving a better weight to first sentences, relying on the fact that introductions bear important information [13],[12]. Then a 'thematic' distance is calculated between the two barycenters, and is considered as the window *central sentence transition score* (figure 1). It is computed according to the augmented concordance distance formula defined in next paragraph.

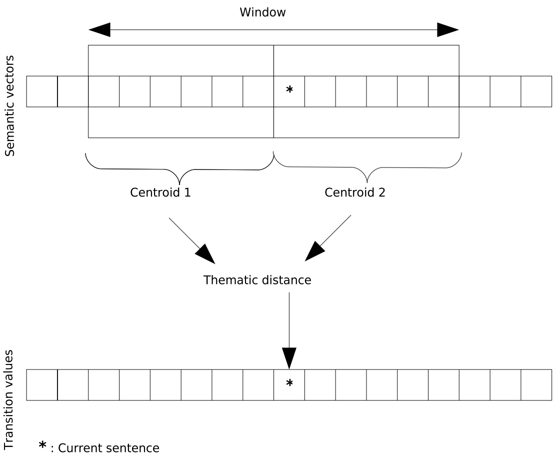


Fig. 1. The transition score of a sentence represent its likelihood of being the first sentence of a segment

*b) Concordance distance:* Semantic vectors resulting from parsing and semantic calculus have $873$ components and most of which having null values. Therefore, either cosine or plain angular distance are not able alone to finely detect a shift in direction. The goal of the concordance distance is to be more discriminant by considering vectors components ranks as well as their values. For the purpose of being more discriminating, we developed the concordance distance, which is itself based based on the concordance measure presented in [5].

Considering $\vec{A}$ and $\vec{B}$ two semantic vector representing respectively two sentences $A$ and $B$. Their values are sorted from the most activated to the least activated and only $\frac{1}{3}$ of the original vectors is kept. $\vec{A_{sr}}$ and $\vec{B_{sr}}$ are respectively the sorted and reduced versions of $\vec{A}$ and $\vec{B}$.

Obviously, if both vectors have no common components then their distance is set to 1. If $\vec{A_{sr}}$ and $\vec{B_{sr}}$ have common components, two differences are necessary to evaluate their 'distance':

— THE RANK DIFFERENCE: if $i$ is the rank of $C_t$ a component of $\vec{A_{sr}}$ and $\rho(i)$ the rank of the same component in $\vec{B_{sr}}$, the rank difference is calculated as:

$$E_{i,\rho(i)} = \frac{(i-\rho(i))^2}{Nb^2 + (1+\frac{i}{2})} \tag{1}$$

Where $Nb$ is the number of values kept in the sorted vector.

— THE INTENSITY DIFFERENCE: One has to compare the intensity of common strong components. If $a_i$ is the intensity of $i$ rank component from $\vec{A_{sr}}$ and $b_{\rho(i)}$ the intensity of the same component in $\vec{B_{sr}}$ (its rank is $\rho(i)$), then intensity difference is given by the formula:

$$I_{i,\rho(i)} = \frac{\left\| a_i - b_{\rho(i)} \right\|}{Nb^2 + (\frac{1+i}{2})} \tag{2}$$

These two differences allow us to compute an intermediate value $P$:

$$P(\vec{A_{sr}}, \vec{B_{sr}}) = \left(\frac{\sum_{i=0}^{Nb-1} \frac{1}{1+E_{i,\rho(i)} * I_{i,\rho(i)}}}{Nb}\right)^2 \tag{3}$$

As $P$ concentrates on components intensities and ranks, the overall components direction is introduced by mixing $P$ with the classical vector angular distance. If $\delta(\vec{A}, \vec{B})$ is the angular distance between $\vec{A}$ and $\vec{B}$, then:

$$\Delta(\vec{A_{sr}}, \vec{B_{sr}}) = \frac{P(\vec{A_{sr}}, \vec{B_{sr}}) * \delta(\vec{A}, \vec{B})}{\beta * P(\vec{A_{sr}}, \vec{B_{sr}}) + (1-\beta) * \delta(\vec{A}, \vec{B})} \tag{4}$$

Where $\beta$ is a coefficient used to give more weight (or less) to $P$. $\Delta(\vec{A_{sr}}, \vec{B_{sr}})$ is the *concordance value*, presented in [4]. It is easy to prove that neither $P$ nor $\Delta(\vec{A_{sr}}, \vec{B_{sr}})$ are symmetric. But $Transeg$ needs a symmetric value (a distance). To have one, we just have to compute an average between $\Delta(\vec{A_{sr}}, \vec{B_{sr}})$ and $\Delta(\vec{B_{sr}}, \vec{A_{sr}})$:

$$D(\vec{A}, \vec{B}) = \frac{\Delta(\vec{A_{sr}}, \vec{B_{sr}}) + \Delta(\vec{B_{sr}}, \vec{A_{sr}})}{2} \tag{5}$$

*c) Practical example of the concordance distance: :* If we considerate these two sentences:

- "Car overuse has a disastrous impact on the environment and more precisely on the global warming."
- "Our new car model only emit $127g$ of $CO_2$ per miles, be kind to the environment, buy a X car !"

Both sentences speak about environment and cars, but the first sentence is about ecology, the second sentence is an advertisement for a car. If we represent both sentences with semantic vectors:

- $\vec{Ph_1} = [0,0,0,0,0,1,0,0,0,0,2,0,0,0,0]$ for the first sentence.
- $\vec{Ph_2} = [0,0,0,0,0,2,0,0,0,0,1,0,0,0,0]$ for the second sentence.

Where the sixth value of vectors is the CAR concept and the eleventh the ECOLOGY concept[1]. If we compute a normalized angular distance[2] between the two vectors, we obtain a value of $0,41$. Which means that the to sentences are close enough to be part of the same topic (see next paragraph for the threshold value).

If we compute a concordance distance between the two sentences, the result is $0.56$ (rounded down). Such a value undoubtly differentiate the two sentences.

*d) Transition zones:* Once each sentence has a transition score, parts of the text where boundaries are likely to appear are identified. These zones are successive sentences with a score greater than a determined threshold $S$ (figure 2). Since the ideal transition zone is assumed to be a two sentences long text segment, isolated sentences are ignored. Distance helps

[1] The representation as been greatly simplified for the purpose of the demonstration

[2] The result would be between 0 and 1 instead of 0 and $\frac{\pi}{2}$

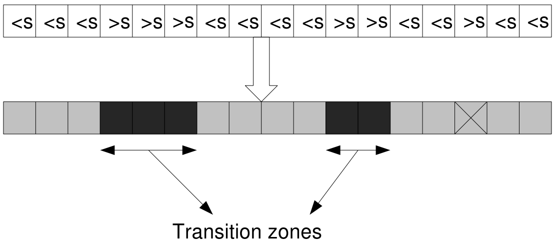


Fig. 2. Identifying transition zones

comparing sentences, but cutting a text into segments needs a maximal distance value acting as a threshold. The chosen one for $D$ (formula 5) is $0.45$, also empirically deduced, like the $10$ sentences segment size. In order to know whether it is corpus dependent or not, $Transeg$ designers browsed two other corpora segmented by human experts, and belonging to the fields of computer science and law (available for the same edition of the DEFT06 competition mentioned before). The threshold seemed to remain constant on these data. This is not a proof that it is completely corpus independent, and needs to be further investigated. However, at a first attempt, it resisted variation, and authors assumed it to be representative of a 'natural trend' of topical discrimination, among other criteria, of course.

The augmented concordance distances were computed between all identified text segments and as a result, the average distance of $0.45$ with a $\sigma$ of $0.08$ appeared to be clearly identified.

*e) Ending sentences and breaking score :* Identifying boundaries inside transition zones needs information about topic ending. The transition score of a sentence is defined as its likelihood of being the first sentence of a segment. The **breaking score** is a sentence likelihood of being *the last sentence of a segment.*

It is obvious that the last sentence of a topic should conclude the topic and more or less introduce the next topic. So the thematic distance of this sentence to its segment should be quite equal to the thematic distance of this sentence to the next segment. The breaking score $B_i$ of the $i$ sentence is:

$$B_i = 1 - |D_p - D_n| \tag{6}$$

Where $D_p$ is the thematic distance of the sentence to the previous segment and $D_n$ the thematic distance of the sentence to the next segment. The closer $D_p$ and $D_n$ are to each other, the closer to $1$ $B_i$ is.

The last step of $Transeg$ method consists in multiplying the transition score of each sentence of a transition zone with the breaking score of the previous sentence. The higher score has high probabilities of being the first sentence of a new segment.

## III. Experiment

The two methods presented in the previous section are obviously quite different, even though tackling the same task and belonging to unsupervised methods. $C99$ concentrates on lexical cohesion to find topic segments by regrouping them. $Transeg$, on the opposite, concentrates on supposed characteristics of topic boundaries to identify them.

To test the incidence of corpus quality and origin on both $c99$ and $Transeg$ results, we used four different corpora during this experiment, impersonating the four variations of our pair of items (clean/raw, natural/artificial).

### *A. Corpus descriptions*

— A 'clean and natural' corpus (Corpus CN). Consisting in $22$ French political discourses extracted from the DEFT'06 training corpus [1]. These discourses have been segmented by human experts into topic segments, and have been cleaned from the noise (typo errors, full capital sentences, etc.). To impersonate this situation, we did not concatenate them. Corpus size: $54,551$ words, $1,895$ sentences.

— A 'raw and natural' corpus (Corpus RN). Consisting in many French political discourses extracted from the same data pool than the previous corpus. Also segmented by human experts, these discourses display lots of noisy items (many typo errors and full capital sentences for example, which in French often introduces diacritics errors). Corpus size: $69,643$ words, $2,214$ sentences.

— A 'clean and artificial' corpus (Corpus CA). Consisting on $134$ concatenated short news from the French news paper "Le Monde". For this corpus each short news is considered as a topic segment. Corpus size: $50,691$ words, $1,574$ sentences.

— A 'raw and artificial' corpus (Corpus RA). Consisting on $131$ concatenated laws extracted from another training corpus from DEFT'06. Corpus size: $53,919$ words, $2,310$ sentences.

As all the cleaning on corpora CN and CA has been hand made, there can still be some noise but far less than before the cleaning. This also explain the relatively short amount of text the experiment was running on. The effort of producing clean data is very heavy. Moreover, other experiments in the domain have been presented with corpora not considerably bigger than ours [7], [20]. In order not to introduce a size bias, we restricted the R corpora (RN and RA) to roughly the same size as the CA and CN corpora (results presented in subsections 4.1 and 4.2). However, since producing raw corpora is very easy, we wanted to know whether size could have an effect on performance results, therefore, we doubled RA and DN corpora sizes, and Tables V (RN: $160,524$ words, $5,445$ sentences) and VI ( RA: $105,350$ words, $4,854$ sentences) have brought up interesting results about the impact of corpus lengths, commented in section 4.3.

### *B. Tolerant measures*

We ran both methods on each corpus and evaluated the results using the DEFT'06 **tolerant recall and precision** described in [1]. They consider as relevant, potential boundary sentences which are in a window around the boundary sentence identified by experts. This evaluation gives a better idea of algorithms efficiency on the task of finding inner texts topic boundaries and does not have a significant influence on the task of finding concatenated texts boundaries. The DEFT'06 organizing committee noticed that the use of either strict or tolerant measures had no effect on the ranking of the submissions they had to evaluate, but gave better scores to all methods.

Note that both methods consider first sentences of texts as a boundary, so every first sentence of the CN and RN corpus texts is counted as a boundary (which means both algorithms have at least one good boundary per text). The results on other corpora are not too affected by this specificity as they are all one big text with several sentences and topic segments.

Even if $c99$ isn't corpus or language sensitive, it has been slightly optimized to English language by using a stop list. As our experiment hab to be the fairest possible, we used $TreeTagger$ to stem the corpora and eliminate tool words from it (based on their categories).

## IV. Results

### A. Clean Vs Raw: The Impact of Corpus Data Quality

The individual results of each of the 22 texts of CN Corpus have been combined in Table I into an average precision and an average recall summarizing the 22 individual values (too long to expose here). Table I presents the best conditions output of both methods. It highlights the differences between the two methods. With its default boundaries detection $c99$ has a better precision, but proposes less solutions and so has a worse recall. On the opposite $Transeg$, by actively searching for boundaries, is more sensitive to variation between sentences. It suggests more solutions than $c99$ and so worsens its precision for a better recall. These 'ideal' conditions clearly demonstrate the strengths and weaknesses of both approaches and give hints about means to improve them. As shown in next paragraphs, results considerably worsen whenever corpus quality drops. As soon as we deteriorate the quality of the corpus (Table II) results drop. At first sight, the natural tendency of $c99$ toward precision seems to be conserved, whereas its recall is dramatically affected (with a $1.54\%$ value, one wonders whether it still has a meaning!). A deeper observation of results indicates that $c99$ brings only $3$ sentences back as a boundaries, including the first one, which is a boundary per se. So its relatively good precision is mostly due to the experiment conditions. $Transeg$ seems to be sturdier: Its precision is almost untouched, but its recall has badly deteriorated. A counterintuitive output: $Transeg$ does better than $c99$ in conditions where we thought that the latter would be the most robust. $Transeg$ is considered to be more sensitive to ill-formed sentences, unknown or misspelled words. It was supposed to be distanced by $c99$ on corrupt data. It seems that it is not the case.

TABLE I
Average precision and recall on the $22$ texts of the 'clean and natural' (CN) corpus

| | C99 | Transeg |
|---|---|---|
| Precision | **53.32**% | 45.49% |
| Recall | 23.12% | **38.76**% |

TABLE II
Precision and recall on the 'raw and natural' (DN) corpus

| | C99 | Transeg |
|---|---|---|
| Precision | 35.14% | **43**% |
| Recall | 1.54% | **9.85**% |

### B. Artificial Corpora: Do Corpus Manipulations impact Methods Results?

TABLE III
Precision and recall on the 'clean and artificial' (CA) corpus

| | C99 | Transeg |
|---|---|---|
| Precision | **30.77**% | 22.3% |
| Recall | 5.03% | **19.5**% |

When coming to an artificial but clean corpus (Table III), we retrieve the original balance between the two methods: A precision oriented output for $c99$ and a recall oriented one for $Transeg$. When compared to Table I, the results range is far worse, and becomes difficult to interpret. $C99$ bad recall (around $5\%$) is quite surprising. One would also have expected the opposite: Concatenating distinct texts would make the segmenting task much easier to a word-based algorithm! The 'worst' conditions case, impersonated by the DA Corpus, have not been matched with the worst results by both methods: Only $Transeg$ seems to be very sensitive to this loss in quality and naturality! When compared with the best case, $c99$ precision is less by $11$ points, whereas $Transeg$ precision looses $35$. On the other hand, recalls are strongly impacted by both data corruption and manipulation (a $2.14\%$ recall for $c99$ and a less than $10\%$ one for $Transeg$ are very bad scores). But the orientation seems to be maintained: $C99$ is still leading in precision, and $Transeg$ in recall. However, with such low values, interpretation is risky. One cannot but acknowledge the impact of data reliability on performances degradation.

TABLE IV
Precision and recall on the 'raw and artificial' (DA) corpus

| | C99 | Transeg |
|---|---|---|
| Precision | **42.86**% | 8.02% |
| Recall | 2.14% | **9.27**% |

### C. Complementary Experiment: The Impact of Size on the D Corpora

The impact of corpora length could be assumed to have two opposite effects.

- Either a biggest data would introduce more corruption cases, and thus worsens results (because of the D aspect)
- Or it would provide both algorithms with more opportunities to detect boundaries 'by chance' and thus augment their performances.

In order to see which of these two assumptions is more likely to be supported, we run the experiments on the doubled RN and RA corpora. Results are summarized in Tables V and VI. When comparing Tables II (simple RN corpus) and V (double RN corpus), we see that both methods 'orientation' is

TABLE V
PRECISION AND RECALL ON THE 'RAW AND NATURAL' (RN) BIGGER CORPUS

| | C99 | Transeg |
|---|---|---|
| Precision | 33.33% | **35.7%** |
| Recall | 0.12% | **20.28%** |

TABLE VI
PRECISION AND RECALL ON THE 'RAW AND ARTIFICIAL' (RA) BIGGER CORPUS

| | C99 | Transeg |
|---|---|---|
| Precision | **15.91%** | 8.54% |
| Recall | 5% | **19.29%** |

maintained: $Transeg$ does better, in both precision and recall, but if its precision has been reduced by $8$ points, its recall has improved by $11$!.It seems that 'chance' retrieved boundaries are rather significant. At the same time since precision drops, the number of corrupt data cases prevent good boundaries to be found. Let us notice that the Recall/Precision ratio is invariant with size. On the other hand, $c99$ recall continues to drop down to incredible values. One cannot risk an interpretation.

Table VI has to be compared to Table IV. The better values in Table VI could be interpreted with a 'canceling' effect provided by size. More data corrupted cases, but also more boundaries to be retrieved by chance. This drives us to conclude that providing a bigger set of data adds more noise to algorithms performances interpretation.

### D. Overall Results

In short, the rather unexpected results obtained in experiments could be summarized by the following statements.

- Data quality (i.e. clean vs dirty) has an impact on results of both methods: Best comparative results in recall and precision are achieved with the CN corpus. The dramatic fall in recall for $c99$ (Tables II, V, VI) is difficult to explain. But, surprisingly, a deterioration in quality seems to favor $Transeg$ over $c99$, provided that the corpora keep their 'natural' origin. This is highly counterintuitive.
- Corpus manipulations (i.e. natural vs artificial) has no impact on both methods orientation (a trend toward precision for $c99$ and one toward recall for $Transeg$), in Tables I, III, IV, VI. Values drop for both methods between the best case (Table I) and the worst case (Table IV), but $Transeg$ is more affected than $c99$.When comparing Table I and Table III, there is a loss of $20$ points in general. So artificial corpora seem to lower results. This is also opposite to our previous intuition, in which a general improvement of $c99$ results was expected.
- The comparison between Tables II and IV shows that it is $Transeg$ which is the most affected by data manipulation (artificially built corpora) when corrupt data is present. Since RA corpora are what most evaluation campaigns provide, then $Transeg$ presents a liability which has to be improved.
- The comparison between Tables II and III, the 'diagonal values' is most interesting. The 'artificiality' of the CA corpus has a discriminant effect on $Transeg$ precision (from $43$ in Table II down to $22$ in Table III), but an improvement on its recall (it is doubled). $C99$ looses the $5$ points in precision that it gains in recall. This confirms that $Transeg$ should not run on artificial corpora.
- Tables V and VI show the impact of corpora size. Recall and precision values improve with size for $Transeg$ but they drop down for $c99$. This means that $Transeg$ will resist better with bigger corpora, however, an artificial corpus is what handicaps it most.
- By choosing to show recall and precision values in call cases, and not an $FScore$ built on their ratio, we hope to have grasped the variations introduced by different corpus quality criteria.

## V. CONCLUSION

The impact of copora constitution on methods evaluation cannot be neglected. The previous experiments and their results tend to show it. What is interesting is that given two quite different but corpus independent methods, they cannot resist a deterioration in corpora. The latter can be introduced by either data corruption or data handling. If one is more apt to stand data corruption ($Transeg$) and the other data handling ($c99$) both give their best when quality is granted. For us this means that: (1) If evaluation campaigns organizers do not test their corpora quality, they already handicap unsupervised methods, which will never achieve spectacular results. Supervised methods will tune to corpus bias and overcome them. However they won't be able to do so well with unlearnt data. (2) If they manipulate their data, they will favor lexical based methods over those which are not. (3) If they don't clean it, they will favor sentence or segment based methods. (4) A big corpus is not necessarily more informative on algorithms capabilities than a smaller one. If data is not of a high quality then chance and noise are more likely to temper with results.

Of course, topic-based segmentation methods should be able to deal with any kind of text. But depending on what we want to evaluate the kind of the corpus could be very important. If the goal is to evaluate a method in a practical applicative context, then any corpus should be used as in real conditions anything could happen. On the opposite, if the goal is to to evaluate the validity of a theory or to the feasibility of a task, the choice of the corpus become important. If we do not want to add the complexity of the corpus properties to the complexity of the task, then we should carefully choose our corpora depending on exactly what we want to evaluate. Otherwise the soundness of our evaluations will be jeopardized.

## REFERENCES

[1] J. Azé, T. Heitz, A. Mela, A. Mezaour, P. Peinl, and M. Roche, "Présentation de deft'06 (defi fouille de textes)," *Proceedings of DEFT'06*, vol. 1, pp. 3–12, 2006.

[2] Y. Bestgen and S. Piérard, "Comment évaluer les algorithmes de segmentation automatiques? essai de construction d'un matriel de référence." *Proceedings of TALN'06*, 2006.

[3] J. Chauché, "Un outil multidimensionnel de l'analyse du discours," *Proceedings of Coling'84*, vol. 1, pp. 11–15, 1984.

[4] J. Chauché and V. Prince, "Classifying texts through natural language parsing and semantic filtering." *In Proceedings of LTC'07, third international Language and Technology Conference*, 2007.

[5] J. Chauché, V. Prince, S. Jaillet, and M. Teisseire, "Classification automatique de textes partir de leur analyse syntaxico-sémantique," *Proceedings of TALN'03*, pp. 55–65, 2003.

[6] F. Y. Y. Choi, "Advances in domain independent linear text segmentation," *Proceedings of NAACL-00*, pp. 26–33, 2000.

[7] F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore, "Latent semantic analysis for text segmentation," *Proceedings of EMNLP*, pp. 109–117, 2001.

[8] M. A. Hearst and C. Plaunt, "Subtopic structuring for full-length document access," *Proceedings of the ACM SIGIR-93 International Conference On Research and Development in Information Retrieval*, pp. 59–68, 1993.

[9] S. Kaufmann, "Cohesion and collocation: using context vectors in text segmentation," in *Proceedings of the 37th annual meeting of the ACL*, 1999, pp. 591–595.

[10] A. Labadié and Chauché, "Segmentation thématique par calcul de distance sémantique," *Proceedings of DEFT'06*, vol. 1, pp. 45–59, 2006.

[11] Larousse, *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Paris: Larousse, 1992.

[12] A. Lelu, C. M., and S. Aubain, "Coopération multiniveau d'approches non-supervises et supervises pour la détection des ruptures thématiques dans les discours présidentiels français," *In Proceedings of DEFT'06*, 2006.

[13] C. Lin and E. Hovy, "Identifying topics by position," in *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP–97)*, 1997, pp. 283–290.

[14] K. McCoy and J. Cheng, "Focus of attention: Constraining what can be said next." *in C. Paris, W. Swartout, and W. Mann, ' Natural Language Generation in Artificial Intelligence and Computational Linguistics'*, 1991.

[15] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Computational Linguistics*, vol. 17, pp. 20–48, 1991.

[16] L. Pevzner and M. Hearst, "A critique and improvement of an evaluation metric for text segmentation," *Computational Linguistics*, pp. 113–125, 2002.

[17] J. M. Ponte and W. B. Croft, "Text segmentation by topic," *European Conference on Digital Libraries*, pp. 113–125, 1997.

[18] V. Prince and A. Labadié, "Text segmentation based on document understanding for information retrieval." *In Proceedings of NLDB'07*, pp. 295–304, 2007.

[19] P. Roget, *Thesaurus of English Words and Phrases*. London: Longman, 1852.

[20] N. Stokes, "Spoken and written news story segmentation using lexical chains," in *Proceedings of NAACL '03*, 2003.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 321–324

ISBN 978-83-60810-14-9
ISSN 1896-7094

# N-gram language models for Polish language. Basic concepts and applications in automatic speech recognition systems.

Bartosz Rapp
Laboratory of Language and Speech Technology
ul. Rubież 46, 61-612 Poznań
email: bartosz.rapp@speechlabs.pl

July 14, 2008

***Abstract*—Usage of language models in automatic speech recognition systems usually give significant quality and certainty improvement of recognition outcomes. On the other hand, wrongly chosen or trained language models can result in serious degradation not only recognition quality but also overall performance of the system. Proper selection of language material, system parameters and representation of the model itself is important task during language models construction process.**

**This paper describes basic aspects of building, evaluating and applying language models for Polish language in automatic speech recognition systems, which are intended to be used by lawyer's chambers, judiciary and law enforcements.**

**Language modeling is a part of project which is still early stage of development and work is ongoing so only some basic concepts and ideas are presented in this paper.**

## I. Introduction

GENERALLY it is known that language models (LM) can be effective support for speech recognition process and rise the recognition quality even by tens of percents. This is why elements of language modeling are included in practically all modern automatic speech recognition systems (ASR). Often collected sound material contains many different kinds of noises and artifacts and the speakers utter words in incorrect or negligent way. Automatic analyses of such speech signal on acoustic models ([6]) level can be insufficient to obtain satisfying recognition correctness and certainty. Information supplied by language models can be invaluable help in this kind of situations. Language models used in ASR systems work in similar way to this in which human brain tries to recognize speech on the higher levels of functioning. In case when not all of the heard words are understandable, we are trying automatically replace these words with others—most probable ones. This replacement is done on our knowledge basis (formerly heard or read sentences etc.). We are simply choosing words in the process of statistical and semantical deduction that are the best fit. Hearing sentence "*Ala **na** kota*" we are very certain that the correct sentence should be "*Ala **ma** kota*". Our choice is driven by semantical knowledge and the fact that the second sentence we have heard so many times in our lifetime and probably this is what author of the message was trying to say. Additionally if we know the global context of whole conversation our deduction can be much more certain. Language models can also be used in similar way to correct mistakes which have been made on acoustic models level. Recognized by sound analysis, words not always will suit to the rest of sentence i.e. grammatically or semantically (although sentence is correct according to the rules of Polish language in fact it can have no real meaning and sense i.e. "*ryba jedzie na rowerze*"). As we can see language models are used to correct mistakes and errors which appear as the result of incorrect acoustic recognition or presence of artifacts in the speech signal.

Many language modeling techniques have been developed and proposed during last years. Some of them are based on "brute-force" statistical analysis (n-gram word LMs and n-gram class LMs ([1]), factored LMs etc.) and others are more formal with heavy theoretical and linguistic background (PCFG or HPSG ([2])). In this article only n-gram language models will be discussed more widely.

N-gram language model is a set of probabilities of the word sentences $P(w_i|w_1,...,w_{i-1})$. These probabilities can be estimated using following equation:

$$P(w_1,...,w_n) = \prod_{i=1}^{n} P(w_i|w_1,...,w_{i-1})$$

Unfortunately chance that out training set will contain the same word sequence $(w_1,...,w_n)$ more than couple of times is rather low. Good solution is therefore to treat the process of word generation as a Markov process (process without memory). According to Markov assumption we accept the fact that only $N$ previous words will have the influence on what word $w_n$) will be. This is what we call n-gram ([1], [3]) language model[1] Selection of the right n-gram length has tremendous impact on the usefulness of language model and depends mainly on what results are expected to be achieved. Higher values of $n$ give better knowledge about context (*discrimination*), lower $n$ values are much more probable to

[1] n-gram means here subsequence of n words from some word sequence. n-gram which has length of 1 are called *unigrams*, 2—*bigrams*, 3—*trigrams*, 4—*tetragrams*, etc.

appearer in the text (*reliability*). In real world application the most common $n$ values are $1 \leq n \leq 4$. It is important to mention that for a vocabulary containing $100\ 000$ words, 4-gram LM can have even $100\ 000^4$ parameters. In fact many of them will represent word sequences that are impossible to appear in real language and can be pruned. It is expected that real language model is usually far less complex than theoretical one. Unfortunately main disadvantage of n-gram language models if fact, it can only estimate probabilities of words from its vocabulary (which is given at the beginning) and adding new word to that vocabulary results in need of rebuilding whole model.

It is needed to remark here, that this project has a research status and the result, that is language models, is only a preliminary version. Main goal of this project (developed at Laboratory of Language and Speech Technology) is to design and implement technology demonstration ASR systems—intended to be used by lawyer's chambers, judiciary and police forces. Basic specifications assume real-time or near to real-time speech recognition of spontaneously dictated speech (i.e. reports, protocols, documents, statements etc.).

## II. Training material selection

Selection of the training texts and language material is one of the most important tasks in LM building process. It should be done with care and some additional factors in mind. First of all training texts need to be appropriate for the occupational category for witch ASR system is intended. Although the same language, there are some slight differences in vocabulary used by lawyers, physicians, politicians and police officers. Each of these groups use some specific therms and sentences typical for the occupation. Nowadays creation of universal language model that suits all kinds of speech is practically impossible. It is worth to mention here that LMs developed as a part of this project are designed with lawyer's chambers, judiciary and police in mind. In the training set following texts are included: many kinds of newspaper texts (for general speech), professional press notes, court protocols, whitens testimonies and statements, police reports and government acts and Parliament speeches. Currently our laboratory owns almost 4GB of different types of texts and linguistic materials. The database is still under development and in near future should reach size of nearly 60GB of data. It should be pointed that while gathering more language material chance to include in this set rarely used words increases. This is why texts which will be used to build language models should be chosen with care.

In case of Polish language which is similar i.e. to Russian ([4]) or Turkish ([5]) vocabulary size should be couple hundred thousands of words. This is only preliminary estimation and further research is needed to obtain the real effective vocabulary size needed for building high quality language model.

## III. Training texts preprocessing

Raw training texts will always contain some mistakes and artifacts like non existing words, misspellings etc. They contain also many unnormalized abbreviation i.e. *mgr inż.* instead of "*magister inżynier*" (*M.Sc.Eng.—Master in Science Engineer*) and numbers in mathematical (i.e. *100* instead of "*sto*") or roman notation. It is obvious that achieving high quality language models will require training materials prepared (preprocessed) in correct way. Having access to word dictionary it is possible to do some automated orthographic and spelling check (manual checking of millions of words is impossible). This is not really necessary step, but if there are possibilities to do this in automated way this kind of spell checking can be applied. It is worth to remark that having really large corpora single and rare misspellings are not harmful during language model construction. If much effort is needed to perform this step it should be skipped. Next normalization of abbreviation is needed. This is the hardest part of preprocessing because shortenings should be expanded to their correct form as should be in sentence according to grammatical and inflection rules and not only to base form i.e. "*Dzisiaj nie ma* ***prof.*** *Jarząbka*" should be "*Dzisiaj nie ma* ***profesora*** *Jarząbka*" and not "*Dzisiaj nie ma* ***profesor*** *Jarząbka*". Generally if we have access to really big corpora it is no need to bother about normalization. Even if one million sentences will be removed from set containing couple billion of sentences this would not be a problem. However some difficulties can be encountered when dealing with small corpora. In this project such relatively small corpora is used to model language specific for lawyers and police. Text material from that set contains large number of numeric values in mathematical notation, abbreviations and shortenings i.e. "KK" instead of "kodeks karny" (penal code) and special characters like i.e. $, §, £, €. In this case normalization is task which should be performed. Normalization is the most complicated part of text preprocessing. Classic approach used while determining the correct word form requires inflectional and semantical analysis of whole sentence. Construction of such automatic analyzer is not an easy task and is not a part of this project. Second proposed approach is to create a set of transformation rules. Using these rules it will be possible to process unnormalized texts and replace all instances of shortenings and numbers, numerals with their expanded and correct form. In this experiment two different approaches will be used to create transformation rule sets. First rules set will be developed manually by linguistic expert according to grammatical and lexical rules of Polish language. As an alternative method some attempts to discover knowledge from collected data will be made. All texts will be divided into two parts. First (test set) will only contain sentences with unnormalized numbers, numerals, shortenings and abbreviations. Second (training set) will consist of sentences containing normalized numbers and numerals together with words (expanded forms) known from abbreviation and shortenings dictionary. Then knowledge discovery algorithms will be used to process sentences from second subset of our corpora on raw and POS labeled sentences. Discovered rules will be applied to the test set. Results of normalization made with manually developed transformation rules set and those discovered from examples will be compared. It is hard to evaluate which approach will be better in practical application. At this moment no goodness index has been proposed to be used to decide which rule set gives better results. At last all words should be capitalized. This will make further text processing much easier and less ambiguous i.e. same words because of capitalized letters will not be classified as different ones: "***Pies** jest najlepszym przyjacielem człowieka*" and "*Moim najlepszym przyjacielem jest **pies** Azor*". Language material preprocessed in this way is ready to be used to train language models.

## IV. Language models

Having decent training material it is possible to proceed with language model construction. At this stage it is needed to recall that the generated language model is static structure and all vocabulary modifications, such as adding new text to training set, require rebuilding whole LM.

The most simple and common form of LM is a technique applying to evaluate probabilities of n-grams sequences for single words. Words, which have not been seen in training material are marked as abstract word *unk*. N-gram models for single words are rather rarely seen in practical applications, because of their requirements for memory and processing power, which result in higher LM's answer time latencies. This kind of resource demand can be unacceptable in real world and real-time application.

One among the methods designed to deal with limitations like such mentioned above is algorithm assigning words to some abstract classes on training texts statistical analysis basis (word exchange algorithm) or using some custom classification function i.e. assigning words to their real part of speech class (POS).

In this project at the early stage of work both mentioned methods are planned to be applied. Statistical derivation of abstract equivalent classes will be made using word exchange algorithm implemented in HTK toolkit. Grammatical POS classification will be made by tagging words according to Inflectional Vocabulary for Polish developed under management of Wiesław Lubaszewski ([8] and morphological analyzer *Morfeusz SIAT* designed by Zygmunt Saloni and Marcin Wolski ([9]). It is expected to achieve better results using inflectional vocabulary. With inflectional vocabulary words can be assigned to 60 POS categories. Furthermore detailed information about word form is also available so in practical applications number of tags can be significantly expanded. In case of *Morfeusz SIAT* there is less than 20 categories. Morphological analyzer contains on the other hand some additional information which are unavailable in inflectional vocabulary. As a part of this project results of POS tagging by these to tools will be compared. Unfortunately *Morfeusz SIAT* and inflectional vocabulary have nowadays some limitations mainly because of small word list. Problem arises in case of proper names, which are commonly seen in training texts and language material.

## V. Probabilities estimation

Because of limited size of language material used to train the language model, some previously unseen and unknown words, will sometimes appear at LM input. Language model is therefore only some approximation of real language. Fact that some of the mentioned word have not been included in the training material does not mean that these words are incorrect on should be considered as non-existent. Occurrences of such words cannot be estimated as a 0 probability. During language model training process some probability mass is needed to be "reserved" for the event of seeing such unknown words. One of the available techniques designed to deal with this problem is discount coefficient factor. There are several algorithms used to calculate discount coefficients form which most popular are: Good-Turing algorithm and absolute discounting used i.e. in HTK toolkit.

## VI. Enhancements techniques

Often some enhancement techniques are used to improve language model quality and decrease its complexity. Most commonly used ones are: the first technique prunes language model and cuts off rare words, the other one is a method of smoothing probabilities. According to this algorithm (probability smoothing) probabilities of some rare n-word sequences are replaced by probabilities of their corresponding shorter contexts (n-1-word sequences). This technique used in parallel with discounting coefficient factor reduces complexity of language model.

## VII. Language model quality estimation

Nowadays most often used metric to evaluate the language model goodness or quality is *perplexity*. This measure has its roots in information theory. In language modeling *perplexity* expresses average number of word choices for current context. Lower *perplexity* value, the better language model was build (it makes lower error). To calculate *perplexity* test text and a language model itself is needed. Unfortunately *perplexity* seems not to be the best measure for evaluating language model quality. This is very simple and basic measure witch additionally highly depends on LM application domain and can be only applied to probabilistic models. Because of its weaknesses it is recommended to support it with other quality indicators such as i.e. Out-Of-Vocabulary (OOV) rate. Some more general measures should be used. Different approaches to language model evaluation which extend *perplexity* or constitute other independent quality factors has been discussed in [11].

## VIII. Summary

In this article basic aspects and concepts of language modeling for applications in real world ASR systems have been discussed. Most of the experiments are conducted using HTK toolkit which is reliable, efficient and proven framework for designing and building automatic speech recognition systems. It is expected to achieve better results during further research of algorithms, techniques and methods (such as using some semantic and context knowledge in language model engine, using fuzzy logic and perhaps some derivations of formal languages elements). It is important to realize that nowadays this is still a research project.

Language model intended to be used in real-time ASR systems need to be applicable, that is, there must be some balance between efficiency and performance. It is needed to notice that this balance is really hard to determine and to achieve. According to our preliminary expectations, language model included in ASR system should increase its recognition accuracy even by 20%. Although we realize that designing good language model for languages like Polish, Turkish, Russian or Finish is quite challenging task.

## REFERENCES

[1] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moor, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, Phil Woodland *The HTK Book*, 2006.

[2] Adam Przepiurkowski, Anna Kupść, Małgorzata Marciniak, Agnieszka Myckowiecka *Formaly opis języka polskiego. Teoria i implementacja*, Akadeicka Oficyna Wydawnicza EXIT, Warsaw, 2002

[3] Steve Young, Gerrit Bloothooft *Corpus-based methods in language and speech processing*, Kluwer Academic Publishers, 1997.

[4] Whittaker E. W. D., Woodland P. C. *Language modeling for Russian and English using words and classes*, Computer speech & language, 2003, vol. 17, pp. 87-104

[5] Ciloglu T., Comez M., Sahin S. *Language modeling for Turkish as an agglutinative language*, Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE 12th, 2004, pp. 461-462

[6] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon *Spoken language processing*, Prentice Hall PTR, 2001.

[7] Barbara Lewandowska-Tomaszczyk *Podstawy językoznawstwa korpusowego*, Wydawnictwo Uniwersytetu Łudzkiego, 2005.

[8] Wiesław Lubaszewski, Henryk Wróbel, Marek Gajęcki, Barbara Moskal, Alicja Orzechowska, Paweł Pietras, Piotr Pisarek, Teresa Rokicka *Słownik fleksyjny języka polskiego*, Wydawnictwa Prawnicze LexisNexis, 2001, ISBN 83-7334-055-6

[9] Marcin Wolski *System znaczników morfosyntaktycznych w korpusie IPI PAN* POLONICA XII, PL ISSN 0137-9712, 2004

[10] Ying Liu, Xiaoyan Zhu *An Efficient Approach of Language Model Applying in ASR Systems*, International Journal of Information Technology, Vol. 11, No. 7, 2005

[11] Stanley Chen, Douglas Beeferman, Ronald Rosenfeld *Evaluation metrics for language models*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 325–329

ISBN 978-83-60810-14-9
ISSN 1896-7094

# SemCAPTCHA—user-friendly alternative for OCR-based CAPTCHA systems

Paweł Łupkowski, Mariusz Urbański
Chair of Logic and Cognitive Science
Institute of Psychology
Adam Mickiewicz University
Szamarzewskiego 89
60-568 Poznań, Poland
Email: {Pawel.Lupkowski, Mariusz.Urbanski}@amu.edu.pl

***Abstract*—In this paper we present a new CAPTCHA system (*Completely Automated Turing Test To Tell Computers and Humans Apart*). This proposal, SemCAPTCHA, is motivated by an increasing number of broken OCR-based CAPTCHA systems and it is based not only on text recognition but also on text understanding.**

**We describe SemCAPTCHA from both user's perspective and system's perspective and compare it to some currently popular CAPTCHAs. We also briefly describe an experiment carried out to test our CAPTCHA on human users.**

## I. Introduction

IN MANY domains there is an increasing demand for simple and efficient way to differentiate real human users from malicious programs (bots). Just a few examples of such domains are: services offering free e-mail accounts, community portals, online polls etc.

One of the most popular ways to tell human users and bot users apart are so called CAPTCHA systems (this acronym stands for *Completely Automated Turing Test To Tell Computers and Humans Apart*).

Design of an effective CAPTCHA system is a difficult task, since two distant needs must be satisfied: it has to be really hard for a machine and at the same moment it has to be simple and friendly for a human. User friendliness is important as CAPTCHAS cannot engage to much of a user cognitive resources and cannot consume to much of her time. Registering a free e-mail account is a good example here. There are many alternative providers of such accounts on the market, so if you want a potential user to solve CAPTCHA on your site, it has to be as unproblematic for her as possible (and you want her to solve it in order to prove that she is a human, not a bot who will send tons of spam from your servers). If a potential user gets irritated, she will go away and pick another provider. To make things more difficult, there's also a third factor: a CAPTCHA has to be open, that is, the algorithms used by a system must be public. The idea is that CAPTCHA efectiveness should be based on hardness of an underlying AI problem and not on a secret cryptographic mechanism or other copyrighted mystery. Finally, test instances of a CAPTCHA should be generated automatically.

Internet users encounter CAPTCHAS very often. Most of them are visual CAPTCHAS where the task consists in recognition of a word or string of symbols (letters, numbers) from a distorted picture. To solve such CAPTCHA a user has to write down words or symbols from the picture. Such systems work e.g. on Yahoo, Gmail, Wirtualna Polska, Gazeta.pl and many other sites. Exemplary CAPTCHAS are presented in table V.

Currently it is an important issue that AI problem underlying such CAPTCHAS is challenged by constantly developing Optical Character Recognition (OCR) systems with increasing success rate. Mori and Malik [8] describe an attack on a visual CAPTCHA EZ-Gimpy used by Yahoo!, which enjoyed a success rate of 92%. In more difficult case of Gimpy they passed the test 33% of the time. As the authors claim: "with our 33% accuracy, this CAPTCHA would be ineffective in applications such as screening out "bots" since a computer could flood the application with thousands of requests." [8, p. 7]. After all, year 2008 seems to be a really bad year for visual CAPTCHAs: Yahoo! CAPTCHA was hacked again (http://osnews.pl, 21.01.2008), as well as Gmail one (http://osnews.pl, 27.02.2008) and MS Windows Live Hotmail (http://arstechnica.com, 15.04.2008). Many visual CAPTCHAS are broken 'out of the box' by PWNtcha system (see http://libcaca.zoy.org/wiki/PWNtcha—examples of 12 broken CAPTCHAS where success rate is from 49% to 100%).

As a consequence, there is a great need for more secure alternative CAPTCHAS, which are based not only on OCR problem. There are some proposals, like question-based CAPTCHA [7], ARTiFACIAL [12], PIX [1], sound oriented CAPTCHAS [3] etc. In our opinion the current situation offers a great motivation to look for an inspiration for CAPTCHA systems not only in simple sensory processing but in higher levels of human data processing.

## II. SemCAPTCHA system

Our proposal is to base a CAPTCHA system on a combination of an OCR problem and some linguistic task, and to apply the effect of positive semantic priming to strengthen human odds against computers. Everything what is needed to break a simple visual CAPTCHA is an good OCR program. Breaking our system—SemCAPTCHA, where "Sem" stands for

This research was partially supported by AMU Faculty of Social Sciences grant No. WSO/133/2006.

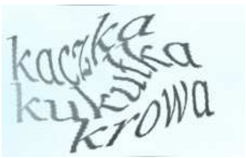


Fig. 1. Sample instance of SemCAPTCHA test

“semantic”—is not that straightforward for a machine and still for a human user it remains quite simple. The process of solving SemCAPTCHA task consists of three steps, based on different cognitive activities (which must be completed within a certain amount of time): reading a text—understanding it—applying user’s knowledge about the world.

### *A. SemCAPTCHA—a user’s perspective*

A SemCAPTCHA test instance consists of a distorted picture, on which three words are presented. All of them are the names of animals. One animal differs from the rest (e.g. it is a mammal among reptiles). The task is to recognize its name and point it by a mouse click. It has to be stressed that the words do not differ substantially as for their graphical properties (like, e.g. length). The difference is of semantic character: one word differs from the other two in its meaning.

An example of such test instance is given in figure 1: a user is presented with the words “kaczka” (a duck), “kukułka” (a cuckoo), “krowa” (a cow; SemCAPTCHA is designed in Polish). The proper answer is “krowa” and the semantic difference is based on taxonomy: ducks and cuckoos are birds while cows are mammals.

To solve this task a user first has to recognize the words from a distorted picture, then identify their meaning and finally find an underlying pattern and the word which does not fit it. The choice of words makes it easy even for not very fluent language users.

In order to make SemCAPTCHA even easier for humans we decided to employ the positive semantic priming effect. Each test instance is preceded by a prime (exposition time is ca. 70 ms). The prime is a word semantically connected with the task solution; in case of the above example it might be a word “mleko” (milk). It is known from cognitive psychology that this setting enables human to recognize a target word much faster than a stand alone target word. Consequently, human user will solve SemCAPTCHA test instances easier and faster (cf. next section, [5] and [6]).

### *B. SemCAPTCHA—a system’s perspective*

SemCAPTCHA is not implemented yet, but the procedures needed for the system are already developed.

SemCAPTCHA works on a word base consisting of 500 animals’ names. Names are grouped in categories, e.g. mammals, birds, reptiles. Each word has its own semantic field (stored as semantic network). Semantic field contains words semantically connected with a given animal name. Each connection of words is marked by a label containing information about relation type and relation strength, expressed by a numerical value 1–100 (as sources for semantic fields generation we used IPI PAN—corpus of Polish developed by the Polish Academy of Sciences—and Google). Such architecture enables efficient and automatic generation of test instances.

To generate a test instance system chooses randomly two categories from the word base. Then it picks (also randomly) one word ($w_1$) from the first category and two words from the second one ($w_2$, $w_3$). Then the system picks a prime for $w_1$, using semantic network stored for $w_1$. The system randomly chooses possible relation strength with $w_1$ (e.g. 50–70) and a word that obeys this restriction. Then a distorted picture is generated using $w_1$, $w_2$, $w_3$ and it is preceded by a prime and a mask.

After a test is generated and displayed SemCAPTCHA starts measuring the time. A solution time (an interval between exposition of a picture and a mouse click) is compared with a standard solution time for SemCAPTCHA. On this basis SemCAPTCHA estimates the probability that a user is a human and decides if a test has been passed or not.

Our experiment shows, that for humans solution time varies from 1,2 to 5,5 seconds (cf. next section, [5] and [6]; more thorough research could help verify these limits). This is one of the most characteristic properties of SemCAPTCHA: it not only generates and scores test instances but it also constantly checks solution time, and its verdict depends not only on correctness of a solution but also on time needed for it. In this point SemCAPTCHA differs substantially from widely used OCR-based CAPTCHA systems.

## III. SemCAPTCHA Experiment

To verify the idea of using linguistic competence and positive semantic priming in SemCAPTCHA system we have carried out an experiment (details on the instruments used and methods of statistical analysis can be found in [5] and are available from the authors).

Our research questions for these issues were:

1) Is the effect of positive semantic priming statistically significant for solution time of SemCAPTCHA test instances?
2) Is the effect of positive semantic priming statistically significant for solution accuracy of SemCAPTCHA test instances?

The experiment consisted of one training task and 10 test instances. A single instance consisted of a picture with 3 Polish words (names of animals). One word was different from the other two in that it was a name of an animal of a different class. For each picture we used one of standard CAPTCHA’s method of distortion. We prepared two sets of tasks, $A$ and $B$, consisting of the same test instances. In an experimental set $A$ each test instance was preceded by a prime, semantically connected with the word which formed the correct solution of a task. A prime was followed by a mask. In a control set $B$ there was no prime. Detailed characteristics of test instances are given in table I.

The sample consisted of 64 students at the Adam Mickiewicz University (19 males, 43 females, 2 no data), who

TABLE I
TASKS CHARACTERISTICS

| Task | Prime (ms) | Mask (ms) | Text dist. | Bg. dist. |
|---|---|---|---|---|
| T1 | 70 | 50 | G-blur | HSV |
| T2 | 60 | 50 | G-blur | RGB |
| T3 | 80 | 50 | G-blur | fog |
| T4 | 90 | 50 | dispersion | HSV |
| T5 | 100 | 60 | dispersion | RGB |
| T6 | 60 | 30 | dispersion | fog |
| T7 | 70 | 50 | Whirl&Pinch | HSV |
| T8 | 70 | 50 | Whirl&Pinch | fog |
| T9 | 70 | 50 | Whirl&Pinch | RGB |
| T10 | 70 | 50 | newspaper printout | HSV |

TABLE II
AVERAGE TIME, ACCURACY AND SUBJECTIVE DIFFICULTY OF TASK SOLUTIONS

| Task | Group | N | Average time (sec.) | Accuracy | Difficulty (Average) |
|---|---|---|---|---|---|
| T1 | $A$ | 31 | 5,5408 | 17 | 6,35 |
|  | $B$ | 33 | 5,9048 | 16 | 6,55 |
| T2 | $A$ | 31 | 2,3467 | 30 | 2,61 |
|  | $B$ | 33 | 2,7859 | 32 | 3,56 |
| T3 | $A$ | 31 | 1,8594 | 27 | 3,26 |
|  | $B$ | 33 | 2,7749 | 31 | 3,48 |
| T4 | $A$ | 31 | 2,7456 | 21 | 5,61 |
|  | $B$ | 33 | 4,7085 | 25 | 6,50 |
| T5 | $A$ | 31 | 1,2047 | 31 | 3,00 |
|  | $B$ | 33 | 3,3308 | 31 | 3,63 |
| T6 | $A$ | 31 | 1,8863 | 30 | 3,03 |
|  | $B$ | 33 | 2,8534 | 32 | 3,47 |
| T7 | $A$ | 31 | 2,5314 | 21 | 4,50 |
|  | $B$ | 33 | 3,5239 | 22 | 5,28 |
| T8 | $A$ | 31 | 1,7810 | 28 | 3,67 |
|  | $B$ | 33 | 3,2051 | 31 | 5,25 |
| T9 | $A$ | 31 | 1,4193 | 30 | 2,67 |
|  | $B$ | 33 | 2,6340 | 32 | 2,97 |
| T10 | $A$ | 31 | 1,5180 | 23 | 3,07 |
|  | $B$ | 33 | 2,6648 | 27 | 3,47 |

volunteered to participate in the experiment. They all belonged to the largest group of Internet users, i.e. people aged between 21 and 25. Participants were randomly distributed over groups $A$ (experimental group) and $B$ (control group).

The subjects were asked to choose on each picture from three names of animals the name of an animal which differs from other two and point it by a mouse click. Solution time was measured as an interval between exposition of a picture and a click. Time and correctness of a solution were written down automatically by a server. After completion of all ten test instances the subjects were asked to fill a short questionnaire concerning subjective difficulty of each task (a complete set of pictures was presented on the monitor at this stage) and their willingness to solve such tasks while surfing the Internet.

The results enable to formulate a positive answer to our first question and a negative answer to the second one (cf. table II). First and foremost, we observed the effect of positive semantic priming in solving test instances of SemCAPTCHA: there was statistically significant difference in time of solving test instances between the experimental group ($A$) and the control group ($B$). Participants from group $A$ solved test instances faster than participants from group $B$ and thus it is possible to differentiate betwen experimental and control group on the basis of the average time of solving test instances. This effect was present in case of eight out of ten test instances (T3 – T10). Lack of positive semantic priming effect in case of the first and second instance can be explained by the need for some practice in solving such tasks.

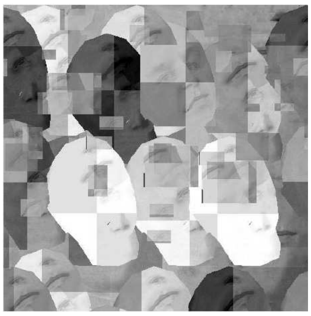


Fig. 2. Example of ARTiFACIAL test [12]

On the other hand, improvement in time of solving test instances does not affect in a statistically significant way the accuracy of solutions. Participants from the experimental group solved test instances just faster, not more accurate than participants from the control group.

## IV. SemCAPTCHA AND OTHER PROPOSALS

As we noticed above, user friendliness is one of the crucial issues for an effective CAPTCHA systems: for humans they should be as easy as possible. Thus it is interesting to compare our system with other CAPTCHAs on the basis of declared subjective difficulty of test instances and declared willingness to use them in practice. For comparison we have chosen CAPTCHAs for which such data were available.

We mentioned already that in our experiment we asked participants to declare subjective difficulty of test instances (on the scale 1–10, where 1 means the simplest). For each test instance subjective difficulty declared by participants from experimental group was slightly lower than the one declared by participants from the control group (however, only in one instance this difference was statistically significant). We observed high correlation between average declared difficulty and average solution time ($r^2 = 0.71$ for group $A$). As a consequence, time of solution seems to be a good estimator of task complexity. This observation gives some base for comparing SemCAPTCHA with other CAPTCHA systems on the objective basis of their solution times.

One of the alternatives for OCR-based CAPTCHA is ARTiFACIAL. It is based on ability to recognize faces. Motivation for this system is similar to ours—make use of higher levels of human data processing. ARTiFACIAL test consists of one picture containing background (with randomly choosen facial features) and a face (exemplary tets instance is presented in figure 2). The task is to find and point six points on such picture (left and right corner of: left eye, right eye and mouth). As could be expected, ARTIFiCIAL is really

TABLE III
AVERAGE TIME (IN SEC.) OF ARTiFACIAL TEST SOLUTION

| task | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| time | 22 | 15 | 16 | 13 | 12 | 11 | 12 | 12 | 11 | 12 |

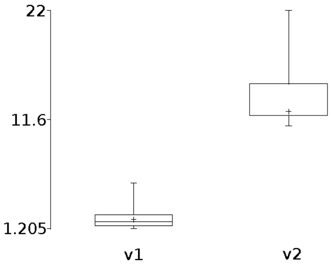


Fig. 3. Average time of solution for SemCAPTCHA (v1) and ARTiFACIAL (v2)

hard for machines, but is it simple enough for human users? ARTiFACIAL authors carried out an experiment on this issue. It consisted of 10 ARTiFACIAL test instances. The sample consisted of 34 subjects (accountants, administrative staff, architects, executives, receptionists, researchers, software developers, support engineers, and patent attorneys). Average solution times are presented in table III (cf. [12, p. 500]).

The mechanics of ARTiFACIAL and SemCAPTCHA are quite similar and it can be claimed that ARTiFACIAL's underlying problem is not more difficult than SemCAPTCHA's one. Thus, if we use the solution time as an estimator of task complexity for human users we may say that ARTiFACIAL is a really complex CAPTCHA system. Average solution time for all tasks is 14 seconds. SemCAPTCHA seems to be much easier, since the average solution time is 2.3 seconds (cf. figure 3).

On the basis of declared willingness to use them in practice we can compare SemCAPTCHA to a simple visual CAPTCHA system—BaffleText. In [4, p. 7] there are given results of a short questionnaire which was ment to investigate BaffleText users feelings about this system. It has been filled by 18 out of 33 subjects (Palo Alto Research Center employees):

1) 16,7 % reported they would be willing to solve a BaffleText every time they sent email;
2) 38,9 % reported they would be willing, if it reduced spam tenfold;
3) 94,4 % reported they would be willing, if it meant those sites had more trustworthy recommendations data;
4) 100 % reported they would be willing to solve one every time they registered for an e-mail account.

In our experiment we asked subjects to answer the same questions (61 out of 64 did this):

1) 15,6 % reported they would be willing to solve a SemCAPTCHA every time they sent email;
2) 43,8 % reported they would be willing, if it reduced spam tenfold;
3) 65,6 % reported they would be willing, if it meant those sites had more trustworthy recommendations data;

TABLE IV
OCR TESTS FOR SEMCAPTCHA

| GOCR | | Asprise OCR | | ABBYY FR | |
|---|---|---|---|---|---|
| words | letters | words | letters | words | letters |
| 0 % | 4,11 % | 0 % | 6,16 % | 13,33 % | 13,01 % |

TABLE V
EXEMPLARY TASKS OF CAPTCHAS USED BY YAHOO!, WP.PL AND GAZETA.PL

| | | |
|---|---|---|
| Yahoo! | wp.pl | gazeta.pl |

4) 34,4 % reported they would be willing to solve one every time they registered for an e-mail account.

We think that this results are very promising for SemCAPTCHA. One possible explanation of low results for third and fourth question is that our subjects were students. They might be not so keen in web security issues as PARC employees.

We have also performed some OCR tests, to see how hard are SemCAPTCHA tests for OCR programs. SemCAPTCHA uses slightly distorted pictures, so we intended to compare them with OCR-based CAPTCHAs currently used on popular portals. We tested our experimental test instances against three OCR programs: GOCR, Asprise OCR and ABBYY Fine Reader 9.0 PE. Results (percentage of correctly recognized words and symbols) are presented in table 4.

For comparison we also performed OCR tests (against the same three programs) for other popular visual CAPTCHAs: the ones used by Yahoo!, wp.pl and gazeta.pl (10 instances for each). These CAPTCHAs do not use regular words, but only strings of symbols (letters and numbers). Exemplary tasks are presented in table V.

For CAPTCHA used by Yahoo! (considered as hard) GOCR recognised 2.82% signs; Asprise OCR 1.41% and ABBYY FR 19.72%. As for wp.pl results were following: GOCR 52.94%, Asprise OCR 16.67%, ABBYY FR 5%. And for gazeta.pl: GOCR 45%, Asprise OCR 0%, ABBYY FR 47.06%. All results are presented in figure 4.

All tested CAPTCHAs are based on an OCR problem. SemCAPTCHA results are comparable with the others (and it

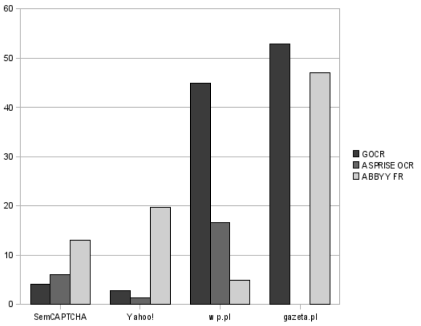


Fig. 4. OCR tests results (in % of recognised symbols)

should be stressed that recognising words in SemCAPTCHA task is only a first step towards solution; cf. section II). Thus we may conclude, that OCR-hardness of SemCAPTCHA is set high enough, i.e. it is at least as hard for machines as CAPTCHAs currently used and still quite easy for human users.

## V. Conclusions

SemCAPTCHA, based on a combination of an OCR problem, some linguistic task and positive semantic priming, seems to be a promising system for telling humans and computers apart. On the one hand, engagement of higher level human data processing makes it harder for machines than currently used visual CAPTCHAs. On the other hand, it is not as complex for human users as other alternatives to current systems. SemCAPTCHA has a simple and open algorithm, is easy for humans and can be designed for any language.

## References

[1] Ahn L., Blum M., Hopper N. J., Langford J. CAPTCHA: Using Hard AI Problems For Security. Retrieved October 11, 2007 from http://www.captcha.net.

[2] Ahn L., Blum M., Langford J. Telling Humans and Computers Apart Automatically. How Lazy Cryptographers do AI. Retrieved October 11, 2007 from http://www.captcha.net.

[3] Chan N. Sound oriented CAPTCHA. Retrieved October 11, 2007 from http://www.captcha.net.

[4] Chew M, Baird H. S. (2003). BaffleText: a Human Interactive Proof. Proceedings of the SPIE/IS&T Document Recognition and Retrieval Conf. X. Santa Clara, CA.

[5] Łupkowski P., Urbański M. (2006). Positive semantic priming as an optimalization tool for automated user authorization systems. Research report, Institute of Psychology, Adam Mickiewicz University (in Polish).

[6] Łupkowski P., Urbański M. (2008). SemCAPTCHA. Telling Computers and Humans Apart by Means of Linguistic Competence and Positive Semantic Priming, In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, J. Zurada (Eds.), Computational Intelligence: Methods and Applications (pp. 525–531). Academic Publishing House EXIT.

[7] Shirali-Shahreza M., Shirali-Shahreza J. (2007). Question-Based CAPTCHA, Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)—Volume 04 (pp. 54–58). IEEE Computer Society, Washington DC.

[8] Mori G., Malik, J. (2003). Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2003. Retrieved October 11, 2007 from http://www.cs.sfu.ca/$\sim$mori/research

[9] Naor M. (1996). Verification of a human in the loop or Identification via The Turing Test. http://www.wisdom.weizmann.ac.il/$\sim$naor/PAPERS/human.ps

[10] Neely J. H. (1991). Semantic priming effects in visual word recognition: A selective review of current findings and theories. In D. Besner, & G. W. Humphreys (Eds.), Basic processes in reading (pp. 264–336). Hillsdale, NJ: Lawrence Erlbaum Associates.

[11] Plaut D. C. (1995). Semantic and Associative Priming in a Distributed Attractor Network. In Proceedings of the 17th Annual Conference of the Cognitive Science Society (pp. 37–42). Hillsdale, NJ: Lawrence Erlbaum Associates.

[12] Rui Y., Liu Z. (2004). ARTiFACIAL: Automated Reverse Turing test using FACIAL features. *Multimedia System* **9**: 493–502.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 331 – 338

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Arabic/English Word Translation Disambiguation Approach based on Naive Bayesian Classifier

Farag Ahmed
Data and Knowledge Engineering Group
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
39106 Magdeburg, Germany
Email: farag.ahmed@ovgu.de

Andreas Nürnberger
Data and Knowledge Engineering Group
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
39106 Magdeburg, Germany
Email: andreas.nuernberger@ovgu.de

***Abstract*—We present a word sense disambiguation approach with application in machine translation from Arabic to English. The approach consists of two main steps: First, a natural language processing method that deals with the rich morphology of Arabic language and second, the translation including word sense disambiguation. The main innovative features of this approach are the adaptation of the Naïve Bayesian approach with new features to consider the Arabic language properties and the exploitation of a large parallel corpus to find the correct sense based on its cohesion with words in the training corpus. We expect that the resulting system will overcome the problem of the absence of the vowel signs, which is the main reason for the translation ambiguity between Arabic and other languages.**

## I. Introduction

INITIALLY, online documents were used predominately by English speakers. Nowadays more than half (50.4%)[1] of web users speak a native language other than English. Therefore, it has become more important that documents of different languages and cultures are retrieved by web search engines in response to the user's request. Cross Language Information Retrieval CLIR allows the user to submit the query in one language and retrieve the results in different languages, providing an important capability that can help to meet that challenge. Cross-Language In-formation Retrieval (CLIR) approaches are typically divided into two main categories: approaches that exploit explicit representations of translation knowledge such as bilingual dictionaries or machine translation (MT) and approaches that extract useful translation knowledge from comparable or parallel corpora.

In the last few years, Arabic has become the major focus of many machine translation projects. Many rich resources are now available for Arabic. For example a GigaWord Arabic corpora, Arabic/English Parallel corpus, which contains several thousands sentence pairs of bilingual text for Arabic and English. The existence of these resources was a crucial factor in building effective translation tools. Bilingual dictionaries (Arabic with other languages) have been used in several Arabic CLIR experiments. However, bilingual dictionaries sometimes provide multiple translations for the same word, which need to be disambiguated. This is due to the fact, that the dictionary may have poor coverage; and it is difficult to select the correct sense of the translated word among all the translations provided by the dictionary.

This paper proposes a method to disambiguate the user translated query in order to determine the correct word translations of the given query terms by exploiting a large bilingual corpus and statistical co-occurrence. The Arabic language properties that hinder the correct match are taken into account by bridging the inflectional morphology gap for Arabic. We use one of the well-known Arabic morphological Analyzers [1] that includes the araMorph package, which we use to translate the user query from Arabic to the English language in order to obtain the sense inventory for each of the ambiguous user query terms.

### *A. Arabic language*

Arabic is a Semitic language, consisting of 28 letters, and its basic feature is that most of its words are built up from, and can be analyzed down to common roots. The exceptions to this rule are common nouns and particles. Arabic is a highly inflectional language with 85% of words derived from tri-lateral roots. Nouns and verbs are derived from a closed set of around 10,000 roots [4]. Arabic has three genders, feminine, masculine, and neuter; and three numbers, singular, dual (represents 2 things), and plural. The specific characteristics of Arabic morphology make Arabic language particularly difficult for developing natural language processing methods for information retrieval. One of the main problems in retrieving Arabic language text is the variation in word forms, for example the Arabic word "kateb" (author) is built up from the root "ktb" (write). Prefixes and suffixes can be added to the words that have been built up from roots to add number or gender, for example adding the Arabic suffix ”ان “ (an) to the word "kateb" (author) will lead to the word "kateban" (authors) which represent dual masculine. What makes Arabic complicated to process is that Arabic nouns and verbs are heavily prefixed. The definite article ”ال “ (al) is always attached to nouns, and many conjunctions and prepositions are also attached as prefixes to nouns and verbs, hindering the retrieval of morphological variants of words [5]. Arabic is different from English and other Indo-European languages with respect to a number of important aspects. Words are written from right to left. It is mainly a consonantal language in its written forms, i.e. it excludes vowels. Its two main parts of speech are the verb and the noun in that word

[1] http://www.worldlingo.com/en/resources/ languag_statistics.html

978-83-60810-14-9/08/$25.00 © 2008 IEEE

order and these consist, for the main part, of trilateral roots (three consonants forming the basis of noun forms that are derived from them). It is a morphologically complex language in that it provides flexibility in word formation: as briefly mentioned above, complex rules govern the creation of morphological variations, making it possible to form hundreds of words from one root [6].

Arabic poses a real translation challenge for many reasons; Arabic sentences are usually long and punctuation has no or little affect on interpretation of the text. Contextual analysis is important in Arabic in order to understand the exact meaning of some words. Characters are sometimes stretched for justified text (word will be spread over a bigger space than other words), which hinders the exact match for the same word. In Arabic, synonyms are very common, for example, “year” has three synonyms in Arabic سنة ، حول ، عام and all are widely used in everyday communication. Despite the previous issues and the complexity of Arabic morphology, which impedes the matching of the Arabic word, another real issue for the Arabic language is the absence of diacritization (sometimes called vocalization or voweling). Diacritization can be defined as a symbol over and underscored letters, which are used to indicate the proper pronunciations as well as for disambiguation purposes. The absence of diacritization in Arabic texts poses a real challenge for Arabic natural language processing as well as for translation, leading to high ambiguity. Though the use of diacritization is extremely important for readability and understanding, diacritization are very rarely used in real life situations. They don’t appear in most printed media in Arabic regions nor on Arabic Internet web sites. They are visible in religious texts such as the Quran, which is fully diacritized in order to prevent misinterpretation. Furthermore, the diacritization are present in children’s books in school for learning purposes. For native speakers, the absence of diacritization is not an issue. They can easily understand the exact meaning of the word from the context, but for inexperienced learners as well as in computer usage, the absence of the diacritization is a real issue. When the texts are unvocalized, it is possible that several words have the same form but different meaning.

### *B. Tim Buckwalter Arabic morphological analyzer (BAMA)*

(BAMA) is the most well known tool for analyzing Arabic texts. It consists of a main database of word forms that interact with other concatenation databases. An Arabic word is considered a concatenation of three regions: a prefix region, a stem region and a suffix region. The prefix and suffix regions can be null. Prefix and suffix lexicon entries cover all possible concatenations of Arabic prefixes and suffixes, respectively. Every word form is entered separately. It takes the stem as the base form and also provides information on the root. (BAMA) morphology reconstructs vowel marks and provides an English glossary. It returns all possible compositions of stems and affixes for a word. (BAMA) groups together stems with a similar meaning and associates it with a lemma ID. The (BAMA) contains 38,600 lemmas. For our work, we use the araMorph package. araMorph is a sophisticated java based Buckwalter analyzer. This package is described in detail in section 1.2.

## II. Word Sense Disambiguation

The meaning of a word may vary significantly according to the context in which it occurs. As a result, it is possible that some words can have multiple meanings. This problem is even more complicated when those words are translated from one language into others. Therefore there is a need to disambiguate the ambiguous words that occur during the translations. The word translation disambiguation, in general, is the process of determining the right sense of an ambiguous word given the context in which the ambiguous word occurs (word sense disambiguation; WSD). We can define the WSD problem, as the association of an occurrence of an ambiguous word with one of it is proper sense. As described in the first section, the absence of the diacritization in most of the Arabic printed media or on the Internet web sites lead to high ambiguity. This makes the probability that the single word can have multiple meaning a lot higher. For example, the Arabic word “يعد“ can have these meanings in English (Promise, Prepare, count, return, bring back) or the Arabic word “علم“ can have these possible meanings (flag, science, he knew, it was known, he taught, he was taught). The task of disambiguation therefore involves two processes: Firstly, identifying all senses for every word relevant, secondly assign the appropriate sense each time this word occurs. For the first step, this can be done using a list of senses for each of the ambiguous words existing in everyday dictionaries. The second step can be done by the analysis of the context in which the ambiguous word occurs, or by the use of an external knowledge source, such as lexical resources as well as a hand-devised source, which provides data useful to assigning the appropriate sense for the ambiguous word. In the WSD task, it is very important to consider the source of the disambiguation information, the way of constructing the rules using this information and the criteria of selecting the proper sense for the ambiguous word, using these rules. WSD is considered an important research problem and is assumed to be helpful for many applications such as machine translation (MT) and information retrieval. Approaches for WSD can be classified into three categorizations: supervised learning, unsupervised learning, and combinations of them.

### *A. Word Sense Disambiguation Approaches*

Several methods for word sense disambiguation using a supervised learning technique have been proposed. For example, Naïve Bayesian [7], Decision List [8], Nearest Neighbor [9], Transformation Based Learning [10], Winnow [11], Boosting [12], and Naïve Bayesian Ensemble [13]. Using bilingual corpora to disambiguate words is leveraged by [14]. For all of these approaches, the one using Naïve Bayesian Ensemble is reported as the best performance for word sense disambiguation tasks with respect to the data set used [13]. The idea behind the previous approaches is that it is nearly always possible to determine the sense of the ambiguous word by considering its context, and thus all methods attempt to build a classifier, using features that represent the context of the ambiguous word. In addition to supervised approaches for word sense disambiguation, unsupervised approaches and combinations of them have been also proposed for the same purpose. For example, [15] proposed an Automatic word sense discrimination which divides the occurrences of a word into a number of classes by determining for any two occurrences whether they belong to the same sense or not, which is then used for the full word sense disambiguation task. Examples of unsupervised approaches were proposed in [16][17][18][19][20][21]. [22] an unsupervised learning method using the Expectation-Maximization (EM) algorithm for text classification problems, which then was improved in [23] in order to apply it to the WSD problem. In [24] the combination of both supervised and unsupervised lexical knowledge methods for word sense disambiguation have been studied. In [25] and [26] rule-learning and neural networks have been used respectively.

Corpora based methods for word sense disambiguation has also been studied. Corpora based methods provide an alternative solution for overcoming the lexical acquisition bottleneck, by gathering information directly from textual data. Due to the expense of manual acquisition of lexical and disambiguation information, where all necessary information for disambiguation have to be manually provided, supervised approaches suffer from major limitations in their reliance on pre-defined knowledge sources, which affects their inability to handle large vocabulary in a wide variety of contexts. In the last few years, the natural data in electronic form has been increased, which helps the WSD researches to extend the coverage of the existing system or train a new system. For example, in [27] and [28] the usage of parallel, aligned Hansard Corpus of Canadian Parliamentary debates for WSD has been performed, in [29] the authors use monolingual corpora of Hebrew and German for WSD. All of the previous studies were based on the assumption that the mapping between words and word senses is widely different from one language to another. Unlike machine translation and dictionaries, parallel corpora provide very high quality translation equivalents that have been produced by experienced translators, who associate the proper sense of a word based on the context that the ambiguous word is used in.

In the next section, we describe the proposed algorithm based on Naïve Bayesian classification, explaining the way of solving or at least relaxing the Arabic morphological issues. Afterward, we explain the features used to represent the context in which ambiguous words occur, followed by experimental results, which show the results of disambiguating some ambiguous words using a parallel corpus. This paper closes with a conclusion and future work.

## III. Proposed Approach

Our approach is based on exploiting parallel texts in order to find the correct sense for the translated user query term. The minimum query length that the proposed approach accepts is two. Given the user query, the system begins by translating the query terms using the araMorph package. In case the system suggests more than one translation (senses inventory) for each of the query terms, the system then starts the disambiguation process to select the correct sense for the translated query terms. The disambiguation process starts by exploiting the parallel corpus, in which the Arabic version of the translation sentences matches fragments in the user query. A matched fragment must contain at least one word in the user query beside the ambiguous one. The words could be represented in surface form or in one of its variant forms. Therefore, and to increase the matching score quality, special similarity score measures will be applied in order to detect all word form variants in the translation sentences in the training corpus.

### *A. Bridging the Inflectional morphology gap*

The rich inflectional morphology languages face a challenge for machine translation systems. As it is not possible to include all word form variants in the dictionaries, inflected forms of words for those languages contain information that is not relevant for translation. The inflectional morphology differences between high inflectional language and poor inflectional language, presents a number of issues for the translation system as well as to disambiguation algorithms. This inflection gap causes a matching challenge when translating between rich inflectional morphology and relatively poor inflectional morphology language. It is possible to have the word in one form in the source language, while having the same word in few forms in the target language. This causes several issues for word translation disambiguation, where more unknown words forms will exist in the training data and will not be recognized as a relevant to the user query terms.  As a result, it is possible to have lower matching scores for those words, even though there is a high occurrence of them in the training data.

The aim of this initial step is to alleviate the Arabic language morphology issues, which has to be done before accessing the Arabic language by the disambiguation algorithm. In order to deal with Arabic morphology issues, we used araMorph package [1] , which is based on an n-gram model. The two main benefits of using the n-gram approach are that it is language independent and that it takes the misspelled and transliteration words that we face in the training data into account. The approach differs from the existing Arabic approaches done in this manner, in terms of the enhancement of the pure n-gram model. Based on the conflation approach, in this step, all word form variants of the user query and in the training data will be detected based on similarity scores measured between the user query term and the words existing in the lexicon.

To describe the problem more clearly, we consider for simplicity the Arabic word "دين" as described in section II. The absence of the diacritization from the Arabic printed media or the Internet web sites causes high ambiguity. The Arabic word "دين" has two translations in English (Religion or debt). We calculate the occurrences of this word in the training corpus for both senses. As it is shown in Table I the word "دين " was found in basic form for the sense (Religion) 49 times and for the sense (Debt) only 10 times.

As Table II shows, when we consider the inflectional form for the word "دين " we see that the occurrence of the inflectional form for the word "دين " with the sense (Religion) is 1192 and with the sense (Debt) is 231.

Table III shows sentence examples from the training corpus where the ambiguous word "دين "appears in basic or inflectional form with both senses. Detecting all word forms variants of the user query terms in the corpus is very essential when computing the score of the synonym sets, as it is shown in the Table II. More than 1386 sentences will be visible to the approach to disambiguate the ambiguous word "دين". For more details about the word form variant detection and their impact on the retrieval performance, we refer the reader to our previous work [2][3].

TABLE I.
THE OCCURRENCE OF THE AMBIGUOUS WORD "دين" IN THE BASIC FORM FOR BOTH SENSES

| The ambiguous word | senses | Occurrence in training data |
|---|---|---|
| | | Basic form |
| دين | Religion | 49 |
| دين | Debt | 10 |
| Total | | 59 |

TABLE II.
THE OCCURRENCE OF THE INFLECTIONAL FORM FOR THE AMBIGUOUS WORD "دين" FOR BOTH SENSES

| The ambiguous word | senses | Occurrence in training data |
|---|---|---|
| | | Inflectional form |
| الدين | The Religion | 75 |
| والدين | And the Religion | 22 |
| الأديان | The Religions | 45 |
| والأديان | The Religions | 7 |
| الدينية | The Religious | 63 |
| والدينية | And the Religious | 28 |
| Total | | 240 |
| الدين | The debt | 860 |
| والدين | And the debt | 22 |
| الديون | The debts. | 255 |
| والديون | And the debts. | 9 |
| Total | | 1146 |

In the following, our approach based on the Naïve Bayesian algorithm, where we learn words and their relationships from a parallel corpus, taking into account that the morphological inflection that differs across the source and target languages, is described.

### *B. Approach based on Naïve Bayesian Classifiers (NB)*

The Naïve Bayesian Algorithm was first used for general classification problems. For WSD problems it had been used for the first time in [28]. The approach is based on the assumption that all features representing the problem are conditionally independent giving the value of classification variables. For a word sense disambiguation tasks, giving a word $W$, candidate classification variables $S = (s_1, s_2, ..., s_n)$, which represent the senses of the ambiguous word, and the feature $F = (f_1, f_1, ..., f_1)$ which describe the context in which an ambiguous word occurs, the Naïve Bayesian finds the proper sense $s_i$ for the ambiguous word $W$ by selecting the sense that maximizes the conditional probability of occurring given the context. In other words, NB constructs rules that achieve high discrimination between occurrences of different word-senses by a probabilistic estimation. The Naive Bayesian estimation for the proper sense can be defined as follows:

$$P(s_i \mid f_1, f_2, ...., f_n) = p(s_i) \prod_{j=0}^{m} p(f_j|s_i) \qquad (1)$$

TABLE III.
SENTENCES EXAMPLES FOR THE AMBIGUOUS WORD "دين" FOR BOTH SENSES IN BASIC AND INFLECTIONAL FORM

| sense | Form | Arabic sentence | English translation |
|---|---|---|---|
| Religion | Basic | لأن الإسلام الذي هو *دين* حوار وانفتاح على الناس | because Islam, which is a ***religion*** of dialogue and openness to people |
| Debt | Basic | نضيف إلى ذلك أن الولايات المتحدة الأمريكية أكبر دولة مدينة في العالم، فلديها 400 مليار دولار عجزاً في ميزانيتها، يتم تمويلها عن طريق الاقتراض من المؤسسات الدولية والبنوك أو عن طريق تحويل هذا العجز إلى *دين* في الموازنة | In addition, the USA is the biggest debtorcountry in the world as it has a budget deficit of $400 billion which is financed through borrowing from international institutions and banks or through converting such a deficit into a budget ***debt***. |
| Religion | Infl. | ودعوا الوزير إلى التراجع عن قرار افتتاح المدرسة واستبدالها بمركز ثقافي ينشر تعاليم *الدين* والثقافة العربية | They called on the Minister to backtrack from that decision and to replace that school with a cultural centre promoting tenets of the ***religion*** and Arabic culture. |
| Debt | Infl. | وأكد الوزير أن الدين الخارجي على مصر هو في مستويات آمنة استناداً إلى ترتيبات جدولة *الدين* في نادي باريس | The minister emphasized that the foreign debt on Egypt was at safe levels due to the arrangements ***of debt*** scheduling in Paris Club. |

The sense $s_i$ of a polysemous word $w_{amb}$ in the source language is defined by a synonym set (one or more of its translations) in the target language. The features for WSD, that are useful for identifying the correct sense of the ambiguous words, can be terms such as words or collocations of words. Features are extracted from the parallel corpus in the context of the ambiguous word. The conditional probabilities of the features $F = (f_1, f_2, ..., f_m)$ with observation of sense $s_i$, $P(f_j|s_i)$ and the probability of sense $s_i$, $P(s_i)$ are computed using maximum-likelihood estimates with $P(f_j|s_i) = C(f_j, s_i) / C(s_i)$ and $P(s_i) = C(s_i)/N$. $C(f_j, s_i)$ denotes the number of times feature $f_j$ and sense $s_i$ have been seen together in the training set. $C(s_i)$ denotes the number of occurrences of $s_i$ in the training set and $N$ is the total number of occurrences of the ambiguous word $w_{amb}$ in the training dataset.

### *C. Features Selection*

The selection of an effective representation of the context (features) plays an essential role in WSD. The proposed approach is based on building different classifiers from different subset of features and combinations of them. Those features are obtained from the user query terms (not counting the ambiguous terms), topic context and word inflectional form in the topic context and combinations of them.

In our algorithm, query terms are represented as sets of features on which the learning algorithm is trained. Topic context is represented by a bag of surrounding words in a large context of the ambiguous word:

$$F = \{w_{w_{amb-k}}, \ldots, w_{w_{amb-2}}, w_{w_{amb-1}}, w_{amb}, w_{w_{amb+1}}, w_{w_{amb+2}}, \ldots$$
$$, w_{w_{amb+k}}, q_1, q_2, \ldots, q_n\}$$

where $k$ is the context size, $w_{amb}$ is the ambiguous word and *amb* its position. The ambiguous word and the words in the the context can be replaced by their inflectional forms. These forms and their context can be used as additional features. Thus, we obtain *F'* which contains in addition to the ambiguous word $w_{amb}$ and its context the inflectional forms $w_{\inf_i}$ of the given sense and their context, as it is shown in table II. Detecting all word form variants of the user query terms in the corpus will make 1386 sentences visible to the approach to disambiguate the ambiguous word "دين". In addition, we count for each context word the number of occurrences of this word and all its inflectional forms, i.e.

$$F' = F \mathop{\square}_{i=0}^{l} \{w_{w_{\inf_i -k}}, \ldots, w_{w_{\inf_i -2}}, w_{w_{\inf_i -1}}, w_{\inf_i}, w_{w_{\inf_i +1}}, \ldots, w_{w_{\inf_i +k}}\} .$$

### D. General Overview of the System

As Figure 1 shows, the system starts to process the user query. The input is a natural language query $Q$. The query is then parsed into several words $q_1, q_2, q_3, .., q_n$. Each word is then further processed independent of the other words. Since the dictionary does not consist of all word forms of the translated word, only the root form, for each $q_m$ in our query, we find its morphological root using the araMorph tool[2].

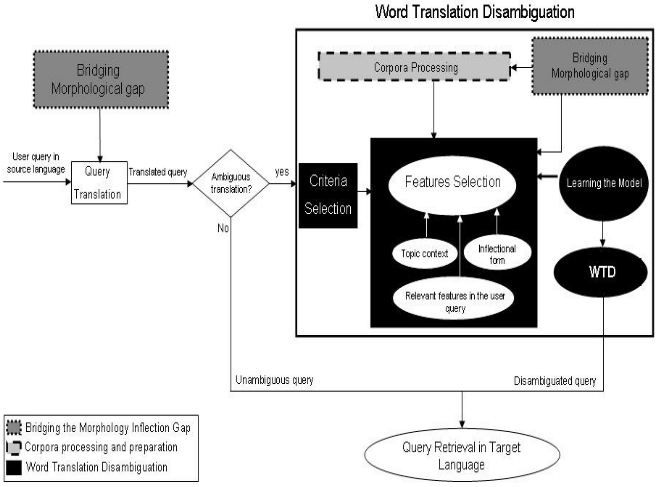

Word Translation Disambiguation
Bridging Morphological gap
Corpora Processing
Bridging Morphological gap
User query in source language
Query Translation
Translated query
Ambiguous translation?
yes
No
Criteria Selection
Features Selection
Learning the Model
Topic context
Inflectional form
Relevant features in the user query
WTD
Unambiguous query
Disambiguated query
Query Retrieval in Target Language
Bridging the Morphology Inflection Gap
Corpora processing and preparation
Word Translation Disambiguation

Fig. 1 General overview of the system

After finding the morphological root of each term in the query, the query term will be translated. In case the query term has more than one translation, the model will provide a list of translations (sense inventory) for each of the ambiguous query terms. Based on the obtained sense inventory for the ambiguous query term, the disambiguation process can be initiated. The algorithm starts by computing the scores of the individual synonym sets. This is done by exploiting the parallel corpora in which the Arabic version of the translated sentences matches words or fragments of the user query, while matched words of the query must map to at least two words that are nearby in the corpus sentence. These words could be represented in surface form or in one of its inflectional forms. Therefore, and to increase the matching score quality, special similarity score measures will be applied in order to detect all word form variants in the translation sentences in the training corpora. Since the Arabic version of the translation sentences in the bilingual corpora matches fragments in the user query, the score of the individual synonym sets can be computed based on the features that represent the context of the ambiguous word. As additional features the words in the topic context can be replaced by their inflectional form. Once we have determined the features, the score of each of the sense sets can be computed. The sense which matches the highest number of features will be considered as the correct sense of the ambiguous query term and then it will be the best sense that describes the meaning of the ambiguous query term in the context.

### E. Illustrative examples

To consider how the algorithm perform the disambiguation steps, consider the following simple query:

رسم جمركي للسلع
(A customs tax of commodities)

Step 1: The natural language query $Q$ is parsed into several words $q_1, q_2, q_3, .., q_n$.

Step 2: For each $q_m$ in the query, we find its morphological root.

Step 3: Translation of the query terms and creation of the sense inventory array in case of any for each of the query term is done. Table IV shows the sense inventory for each of the ambiguous query terms.

Step 4: The disambiguation process is initiated. The algorithm starts by computing the scores of the individual synonym sets:

- Number of times feature $f_j$ and sense $s_i$ have been seen together in the training set is computed.
- Number of occurrences of $s_i$ in the training set is computed.
- The total number $N$ of occurrences of the ambiguous word $w_{amb}$ in the training dataset is computed.
- The disambiguation score is computed and the sense which matches the highest number of features will be considered as the correct sense of the ambiguous query term.

Table V shows the disambiguation scores of the individual synonym sets for each ambiguous query terms with other query terms. As Table V shows there are 135 possible translations set for the original query in source language.

[2] http://www.nongnu.org/aramorph/

TABLE IV.
SENSE INVENTORY FOR EACH OF THE AMBIGUOUS QUERY TERMS

| Original Query term | Sense inventory (Possible English Translations) |
|---|---|
| رسم | [fee, tax, drawing, sketch, illustration, prescribe, trace, sketch, indicate, appoint] |
| جمركي | [customs, tariff, customs, control] |
| للسلع | [crack, rift, commodities, commercial, goods] |

TABLE V.
DISAMBIGUATION SCORES FOR EACH POSSIBLE TRANSLATIONS SETS

| S/N | query | score | S/N | query | score |
|---|---|---|---|---|---|
| 1 | fee AND (customs OR crack) | 0 | 71 | appoint AND (tariff OR crack) | 0 |
| 2 | fee AND (customs OR rift) | 0 | 72 | appoint AND (tariff OR rift) | 0 |
| 3 | fee AND (customs OR commodities) | 0 | 73 | appoint AND (tariff OR commodities) | 0 |
| 4 | fee AND (customs OR commercial) | 0 | 74 | appoint AND (tariff OR commercial) | 0,00058 |
| 5 | fee AND (customs OR goods) | 0 | 75 | appoint AND (tariff OR goods) | 0 |
| 6 | fee AND (control OR crack) | 0 | 76 | trace AND (customs OR crack) | 0 |
| 7 | fee AND (control OR rift) | 0 | 77 | trace AND (customs OR rift) | 0 |
| 8 | fee AND (control OR commodities) | 0 | 78 | trace AND (customs OR commodities) | 0 |
| 9 | fee AND (control OR commercial) | 0 | 79 | trace AND (customs OR commercial) | 0 |
| 10 | fee AND (control OR goods) | 0 | 80 | trace AND (customs OR goods) | 0 |
| 11 | fee AND (tariff OR crack) | 0 | 81 | trace AND (control OR crack) | 0 |
| 12 | fee AND (tariff OR rift) | 0 | 82 | trace AND (control OR rift) | 0 |
| 13 | fee AND (tariff OR commodities) | 0 | 83 | trace AND (control OR commodities) | 0 |
| 14 | fee AND (tariff OR commercial) | 0 | 84 | trace AND (control OR commercial) | 0 |
| 15 | fee AND (tariff OR goods) | 0 | 85 | trace AND (control OR goods) | 0 |
| 16 | tax AND (customs OR crack) | 0,0484 | 86 | trace AND (tariff OR crack) | 0 |
| 17 | tax AND (customs OR rift) | 0,0484 | 87 | trace AND (tariff OR rift) | 0 |
| **18** | **tax AND (customs OR commodities)** | **0,05948** | 88 | trace AND (tariff OR commodities) | 0 |
| 19 | tax AND (customs OR commercial) | 0,05248 | 89 | trace AND (tariff OR commercial) | 0 |
| 20 | tax AND (customs OR goods) | 0,05539 | 90 | trace AND (tariff OR goods) | 0 |
| 21 | tax AND (control OR crack) | 0 | 91 | sketch AND (customs OR crack) | 0 |
| 22 | tax AND (control OR rift) | 0 | 92 | sketch AND (customs OR rift) | 0 |
| 23 | tax AND (control OR commodities) | 0,01224 | 93 | sketch AND (customs OR commodities) | 0 |
| 24 | tax AND (control OR commercial) | 0,00525 | 94 | sketch AND (customs OR commercial) | 0 |
| 25 | tax AND (control OR goods) | 0,01108 | 95 | sketch AND (customs OR goods) | 0 |
| 26 | tax AND (tariff OR crack) | 0,00175 | 96 | sketch AND (control OR crack) | 0 |
| 27 | tax AND (tariff OR rift) | 0,00175 | 97 | sketch AND (control OR rift) | 0 |
| 28 | tax AND (tariff OR commodities) | 0,01399 | 98 | sketch AND (control OR commodities) | 0 |
| 29 | tax AND (tariff OR commercial) | 0,007 | 99 | sketch AND (control OR commercial) | 0 |
| 30 | tax AND (tariff OR goods) | 0,01283 | 100 | sketch AND (control OR goods) | 0 |
| 31 | prescribe AND (customs OR crack) | 0 | 101 | sketch AND (tariff OR crack) | 0 |
| 32 | prescribe AND (customs OR rift) | 0 | 102 | sketch AND (tariff OR rift) | 0 |
| 33 | prescribe AND (customs OR commodities) | 0 | 103 | sketch AND (tariff OR commodities) | 0 |
| 34 | prescribe AND (customs OR commercial) | 0 | 104 | sketch AND (tariff OR commercial) | 0 |
| 35 | prescribe AND (customs OR goods) | 0 | 105 | sketch AND (tariff OR goods) | 0 |
| 36 | prescribe AND (control OR crack) | 0 | 106 | drawing AND (customs OR crack) | 0,00058 |
| 37 | prescribe AND (control OR rift) | 0 | 107 | drawing AND (customs OR rift) | 0,00058 |
| 38 | prescribe AND (control OR commodities) | 0 | 108 | drawing AND (customs OR commodities) | 0,00117 |
| 39 | prescribe AND (control OR commercial) | 0 | 109 | drawing AND (customs OR commercial) | 0,0035 |
| 40 | prescribe AND (control OR goods) | 0 | 110 | drawing AND (customs OR goods) | 0,00058 |
| 41 | prescribe AND (tariff OR crack) | 0 | 111 | drawing AND (control OR crack) | 0,00058 |
| 42 | prescribe AND (tariff OR rift) | 0 | 112 | drawing AND (control OR rift) | 0,00058 |
| 43 | prescribe AND (tariff OR commodities) | 0 | 113 | drawing AND (control OR commodities) | 0,00117 |
| 44 | prescribe AND (tariff OR commercial) | 0 | 114 | drawing AND (control OR commercial) | 0,0035 |
| 45 | prescribe AND (tariff OR goods) | 0 | 115 | drawing AND (control OR goods) | 0,00058 |
| 46 | indicate AND (customs OR crack) | 0 | 116 | drawing AND (tariff OR crack) | 0,00058 |
| 47 | indicate AND (customs OR rift) | 0 | 117 | drawing AND (tariff OR rift) | 0,00058 |
| 48 | indicate AND (customs OR commodities) | 0 | 118 | drawing AND (tariff OR commodities) | 0,00117 |
| 49 | indicate AND (customs OR commercial) | 0 | 119 | drawing AND (tariff OR commercial) | 0,0035 |
| 50 | indicate AND (customs OR goods) | 0,00117 | 120 | drawing AND (tariff OR goods) | 0,00058 |
| 51 | indicate AND (control OR crack) | 0,00058 | 121 | illustration AND (customs OR crack) | 0 |
| 52 | indicate AND (control OR rift) | 0,00058 | 122 | illustration AND (customs OR rift) | 0 |
| 53 | indicate AND (control OR commodities) | 0,00058 | 123 | illustration AND (customs OR commodities) | 0 |
| 54 | indicate AND (control OR commercial) | 0,00058 | 124 | illustration AND (customs OR commercial) | 0 |
| 55 | indicate AND (control OR goods) | 0,00175 | 125 | illustration AND (customs OR goods) | 0 |
| 56 | indicate AND (tariff OR crack) | 0 | 126 | illustration AND (control OR crack) | 0 |
| 57 | indicate AND (tariff OR rift) | 0 | 127 | illustration AND (control OR rift) | 0 |
| 58 | indicate AND (tariff OR commodities) | 0 | 128 | illustration AND (control OR commodities) | 0 |
| 59 | indicate AND (tariff OR commercial) | 0 | 129 | illustration AND (control OR commercial) | 0 |
| 60 | indicate AND (tariff OR goods) | 0,00117 | 130 | illustration AND (control OR goods) | 0 |
| 61 | appoint AND (customs OR crack) | 0 | 131 | illustration AND (tariff OR crack) | 0 |
| 62 | appoint AND (customs OR rift) | 0 | 132 | illustration AND (tariff OR rift) | 0 |
| 63 | appoint AND (customs OR commodities) | 0 | 133 | illustration AND (tariff OR commodities) | 0 |
| 64 | appoint AND (customs OR commercial) | 0,00058 | 134 | illustration AND (tariff OR commercial) | 0 |
| 65 | appoint AND (customs OR goods) | 0 | 135 | illustration AND (tariff OR goods) | 0 |
| 66 | appoint AND (control OR crack) | 0 | | | |
| 67 | appoint AND (control OR rift) | 0 | | | |
| 68 | appoint AND (control OR commodities) | 0 | | | |
| 69 | appoint AND (control OR commercial) | 0,00058 | | | |
| 70 | appoint AND (control OR goods) | 0 | | | |

### *F. Training data*

The proposed algorithm was developed using Arabic/English parallel corpus[3]. This corpus contains Arabic news stories and their English translations. It was collected via Ummah Press Service from January 2001 to September 2004. It totals 8,439 story pairs (Documents), 68,685 sentence pairs, 93,120 segments pairs, 2 Million Arabic words and 2.5 Million English words. The corpus is aligned at sentence level.

## IV. Evaluation

We evaluated our approach through an experiment using the Arabic/English parallel corpus aligned at sentence level. We selected 30 Arabic sentences from the corpus as queries to test the approach. These sentences have various lengths starting from two words up to five words, as future work the maximum query length will be extended. These queries had to contain at least one ambiguous word, which has multiple English translations. In order to enrich the evaluation set, these ambiguous words had to have higher frequencies compared with other words in the training data, ensuring that these words will appear in different contexts in the training data. Furthermore, ambiguous words with high frequency sense were preferred. The sense (multiple translations) of the ambiguous words was obtained from the dictionary. The number of senses per test word ranged from two to nine, and the average was four. For each test word, training data were required by the algorithm to select the proper sense. The algorithm was applied to more than 93,123 parallel sentences. The results of the algorithm were compared with the manually selected sense.

For our evaluation, we built different classifiers from different subsets of features and combinations of them. The first classifier based on features that were obtained from the user query terms and topic context, which was represented by a bag of words in the context of the ambiguous word. The second classifier was based on the topic context and its inflectional form.

In order to evaluate the performance of the different classifiers, we used two measurements: applicability and precision [29]. The applicability is the proportion of the ambiguous words that the algorithm could disambiguate. The precision is the proportion of the corrected disambiguated senses for the ambiguous word. The performance of our approach is summarized in Table IV. The sense, which is proposed by the algorithm was compared to the manually selected sense.

As it is expected the approach is better in the case of long query terms which provide more reach features and worse in short query, especially the one consisting of two words. We consider that the reason for the poor performance is that, when the query consists of few words it is possible that the features which are extracted from the query terms can appear in the context of different senses. For example, consider the query "الدين الإسلامي" (The Islamic religion). When the algorithm goes through the corpus, the ambiguous word "الدين" (The Religion or The debt) will be found in two different context whether in Religion or Debt context. The query term "الإسلامي" (Islamic) can be found in both contexts of the ambiguous word as (Islamic religion) or as a name of bank (Islamic Bank), which is the context of the second sense (Debt). One possible solution for this issue is query expansion. This can be done by exploiting the corpus and suggesting possible term expansion to the user. The user then confirms this term expansion, which will help to disambiguate the ambiguous query term when translating to the target language.

Another reason for the poor performance is that due to the morphological inflectional gap between languages such as Arabic the same word can be found in different forms. In order to increase the performance of the disambiguation process all of these forms need to be detected.

Table VI shows the overall performance of the algorithm based on building two classifiers from different subsets of features and combinations of them. Those features are user query terms, topic context and word inflectional form in topic context and combinations of them. As is shown in Table IV, the performance of the algorithm is poor when using the basic word form. The reason for that, the Arabic word can be represented not just in its basic form, but in many inflectional forms and so we will have more training sentences that will be visible to the algorithm to disambiguate the ambiguous query terms.

Table vi.
the overall performance using Applicability and precision

| classifiers | Applicability | Precision |
|---|---|---|
| Query term + Topic context | 52 % | 68 % |
| Query term+ feature Inflectional form | 82 % | 93 % |

## References

[1] Tim Buckwalter, Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49.

[2] Farag Ahmed and Andreas Nürnberger, N-Grams Conflation Approach for Arabic Text, In: Proceedings of the International Workshop on improving Non English Web Searching (iNEWS 07) In conjunction with the 30th Annual International (ACM SIGIR Conference). Amsterdam City, Netherlands, 2007, pp. 39-46.

[3] Farag Ahmed and Andreas Nürnberger, araSearch: Improving Arabic text retrieval via detection of word form variations, In: Proceedings of the 1st International Conference on Information Systems and Economic Intelligence (SIIE'2008) at Hammamet in Tunisia, 2008, pp. 309-323.

[4] Al-Fedaghi Sabah S. and Fawaz Al-Anzi, Anew algorithm to generate Arabic root-pattern forms. Proceedings of the 11th National Computer Conference, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, 1989, pp. 04-07.

[5] Moukdad, H., Lost in Cyberspace: How do search engines handle Arabic queries? In Access to Information: Technologies, Skills, and Socio-Political Context. Proceedings of the 32nd Annual Conference of the Canadian Association for Information Science, Winnipeg, June 2004, pp. 3-5.

[6] Moukdad, H. and A. Large, Information retrieval from full-text Arabic databases: Can search engines designed for English do the job? Libri 51 (2), 2001, pp. 63-74.

[7] Gale, K. Church, and D. Yarowsky, A Method for Disambiguating Word Senses in a Large Corpus. Computers and Humanities, vol. 26, 1992a, pp. 415-439.

[8] Yarowsky, Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, 1994, pp. 88-95.

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004T18

[9] T. Ng and H. B. Lee, Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, 1996, pp. 40-47.

[10] Mangu and E. Brill, Automatic rule acquisition for spelling correction. In Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 187-194.

[11] R. Golding and D. Roth, A Winnow-Based Approach to Context-Sensitive Spelling Correction. Machine Learning, vol. 34, 1999, pp. 107-130.

[12] Escudero, Gerard, Lluís Màrquez & German Rigau, Boosting applied to word sense disambiguation. Proceedings of the 12th European Conference on Machine Learning (ECML), Barcelona, Spain, 2000, pp. 129-141.

[13] T. Pedersen, A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. In Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics, Seattle, WA, May, 2000, pp. 63–69.

[14] Nancy Ide, N., Parallel translations as sense discriminators. SIGLEX99: Standardizing Lexical Resources, ACL99 Workshop, College Park, Maryland, 1999, pp. 52--61.

[15] Schütze, H.: Automatic word sense discrimination. Computational Linguistics, v.24, n.1, (1998) 97-124.

[16] K. C. Litkowski. 2000. Senseval: The cl research experience. In Computers and the Humanities, 34(1-2), pp. 153-158.

[17] Dekang Lin., Word sense disambiguation with a similarity based smoothed l brary. In Computers and the Humanities: Special Issue on Senseval, 2000, pp. 34:147-152.

[18] Philip Resnik., Selectional preference and sense disambiguation. In Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?, Washington, 1997, pp. 4-5.

[19] David Yarowsky, Word-sense disambiguation using statistical models of Ro-get's categories trained on large corpora. In Proceedings of COL-ING-92, Nantes, France, 1992, pp. 454.460.

[20] Indrajit Bhattacharya, Lise Getoor, Yoshua Bengio: Unsupervised Sense Disambiguation Using Bi-lingual Probabilistic Models. ACL 2004: 287-294.

[21] Hiroyuki Kaji, Yasutsugu Morimoto: Unsupervised Word-Sense Disambiguation Using Bilingual Comparable Corpora. IEICE Transactions 88-D(2), 2005, pp. 289-301.

[22] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell, Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning, 39(2/3), 2000, pp. 103–134.

[23] Hiroyuki Shinnou , Minoru Sasaki, Unsupervised learning of word sense disambiguation rules by estimating an optimum iteration number in the EM algorithm, Proceedings of the seventh conference on Natural language learning at HLT-NAACL, Canada., May 31, 2003, Edmonton, pp. 41-48.

[24] E. Agirre, J. Atserias, L. Padr, and G. Rigau, Combining supervised and unsupervised lexical knowledge methods for word sense disambiguation. In Computers and the Humanities, Special Double Issue on SensEval. Eds. Martha Palmer and Adam Kilgarriff, 2000, pp. 34:1,2.

[25] David Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods. In Meeting of the Association for Computational Linguistics, 1995, pp. 189.196.

[26] Towell and E. Voothees, Disambiguating Highly Ambiguous Words. Computational Linguistics, vol. 24, no. 1, 1998, pp. 125-146.

[27] Brown, P. F., Lai, J. C. & Mercer, R. L., Aligning Sentences in Parallel Corpora, Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics. Berkeley, 1991, pp. 169-176.

[28] Gale, W. A., Church, K. W. & Yarowsky, D., Using bilingual materials to develop word sense disambiguation methods. Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'92), Montréal, 1992, pp. 101-112.

[29] Dagan, Ido and Itai, Alon, Word sense disambiguation using a second language monolingual corpus. Computational Linguistics, 20(4), 1994, pp. 563-596.

[30] Duda, R. O. and Hart, P. E.: Pattern Classification and Scene Analysis, John Wiley, 1973.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Smarty–Extendable Framwork for Bilingual and Multilingual Comprehension Assistants

Todor Arnaudov
Plovdiv University “Paisii Hilendarski”, 24, “Tsar Assen” Str., Plovdiv 4000, Bulgaria
Email: tosh.bg@gmail.com

Ruslan Mitkov
University of Wolverhampton,
Wolverhampton WV1 1SB, UK
Email: r.mitkov@wlv.ac.uk

***Abstract*—This paper discusses a framework for development of bilingual and multilingual comprehension assistants and presents a prototype implementation of an English-Bulgarian comprehension assistant. The framework is based on the application of advanced graphical user interface techniques, WordNet and compatible lexical databases as well as a series of NLP preprocessing tasks, including POS-tagging, lemmatisation, multiword expressions recognition and word sense disambiguation. The aim of this framework is to speed up the process of dictionary look-up, to offer enhanced look-up functionalities and to perform a context-sensitive narrowing-down of the set of translation alternatives proposed to the user.**

## I. Introduction

AT PRESENT, even regular Internet users often access on-line resources in English which require lexical knowledge beyond their current level.

While Machine Translation has been expected to make this problem history, the state-of-the-art is still far from achieving this dream. Full-text machine translation is yet unreliable, and typical users are assisted in translation and language learning with only a variety of word-translation ‘electronic dictionaries’, operating either on-line on the Internet, or as off-line computer software. Generally, such dictionaries offer very simple look-up options and are based on the following functionalities:

1. The u ser types or copy-pastes a word in the input box, or clicks on a word from an alphabetical list of words.

2.The dictionary displays an entry if it contains one whose head word exactly matches the word which is entered. In most cases, the entry is from a scanned version of a paper dictionary. It is not difficult to see that the way a user consults an electronic dictionary is not very different from the way s/he queries paper dictionaries. As with paper dictionaries, the user is presented a list of possible meanings for every word under consideration. In many cases this could cause confusion or misunderstanding. Recent years have seen the development of new lexicographic/language learners’ tools referred to as *comprehension assistants* which seek to enhance the look-up functionality and in particular to narrow down the list of alternative translations through applying basic NLP techniques.

## II. Previous Work

### A. Xerox

The first comprehension assistant reported, *Locolex* (Feldweg and Breidt 1996), was developed by Xerox for French-English and English-German comprehension assistance. Locolex inspired applications developed later including Smarty, which is being discussed in this paper. Locolex, unlike conventional electronic dictionaries, offers the functionality for the user to click on words occurring in any machine-readable text, as opposed to copy-and-pasting separate words. Once the user clicks on a specific word, Locolex performs POS tagging which attempts to identify the correct part-of-speech tag, thus decreasing the number of possible translations. Multiword expressions recognition, based on regular expressions, is also applied, which could help identify the correct translation in particular cases. Locolex also keeps record of user sessions, allowing quick recall of previously checked words.

A recent version of Locolex incorporates word sense disambiguation which contributes to the narrowing down of the set of possible meanings even further.

### *B. Morphologic*

The comprehension assistants developed by Morphologic introduce several additional features.

In particular, one of their products, *MobiMouse* , correctly identifies multiword expressions even if the selected word is not the head of the expression and also offers comprehension assistance in any application running in the operating system environment. The user can click anywhere; the comprehension assistant is running in the background and flashes a translation in the corner of the screen.

### *C. SmartDict*

SmartDict (Kolev, 2005) is an English-Bulgarian dictionary which is somewhere between comprehension assistants and advanced conventional dictionaries. It performs a number of NLP preprocessings, including tokenisation, sentence-splitting, normalisation and multiword expression recognition, but it does not perform reduction of the possible translations by POS-tagging or word-sense disambiguation.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## III. Smarty – framework for bilingual and multilingual comprehension assistants

Inspired by Locolex, we developed Smarty - a framework for comprehension assistants for English-Bulgarian. While Smarty and Locolex share certain similarities, our comprehension assistant has the following distinctive features.

### *A. New Features*

While Smarty and Locolex share certain similarities, our comprehension assistant has the following distinctive features.

*1) Hybrid System*

Smarty represents a hybrid system. The interface is more comprehensive and elaborate than the interface of Locolex or MobiMouse in that it allows users to virtually work with two dictionaries – both an enhanced conventional dictionary and a comprehension assistant (see Fig. 1). In enhanced conventional dictionary mode users can browse freely all dictionary entries and familiarise themselves with the meanings of a specific word. This mode offers additional options such as suffix search, rhyme search, synonymy search etc. which are not present in conventional dictionaries.

*2) New Lexicographical Resource*

WordNet (Miller, 1995) is the lexicographical resource for this comprehension assistant. WordNet adds glosses which can be browsed by the user. Additionally, it makes it possible for word sense disambiguation to be performed.

*3) Extendability*

The alignment between the lexical database of WordNet and corresponding lexical databases for other languages allows bilingual word sense disambiguation to be performed. The incorporation of the existing databases of EuroWordNet (Vossen, 1998) and BalkaNet (Oflazer et. al, 2001), make it perfectly possible for comprehension assistants covering more languages to be developed using the same framework and the same core system. Smarty could be easily extended to be English to Greek, Turkish, Czech, Romanian, Serbian, Italian, Spanish, German, French, Dutch and Estonian comprehension assistants, if their already made lexical databases are available.

### *B. Graphical User Interface*

The aspects of the graphical user interface in the framework, which are different from the framework of conventional dictionaries, are:

*1) Free Text Input*

There is a free text input box, where the full text is pasted or typed. Users can point to the words in their context, instead of copy-pasting (Fig. 2).

*2) Tooltip*

Suggested translated meanings can appear in a tooltip, near the mouse pointer. This is less distracting for users than the translation appearing in a side window.

3) Additional Information

Additional information which assists comprehension is available - glosses, examples of usage etc. and can also be presented to the user in tooltip or in side windows, on demand.

Translation in tooltip proves to be much more convenient for users than translation in separate windows in the same applications or in the worse case - in another application.

POS-tagging allows Smarty to fit the most relevant translations in a tooltip, which could be scanned in few seconds by the user, without touching a scroller and without moving his or her sight away from the context. This also allows immediate continuation of the reading without distraction.

When using a conventional bilingual dictionary, if the queried word has ambiguous part-of-speech and a long entry with sections for each one, the user faces two problems: s/he is forced to figure out the correct part of speech alone; and if the user knows the part-of-speech s/he is forced to scroll and scan with the bare eye where the section for the correct part-of-speech begins. Also, the appearance of the translation in window of another application causes two other time penalties: first, the user is distracted from the reading flow and has to spend time switching attention from the text to the dictionary and back; and second, after the query is done, the user has to find the exact place in the text window where he or she has stopped reading.

### *C. Linguistic Databases*

The framework makes use of at least three linguistics databases: a conventional dictionary database and at least two lexical databases used to provide glosses and word sense disambiguation.

*1) Conventional Dictionary Database*

The conventional dictionary database allows the system to work in conventional dictionary mode. It could be a scan of a paper dictionary, which in this case is to be parsed and transformed in suitable format for processing. This database is used to build indices for *predictive typing* (known also as *autocompletion* ), suffix-search, rhyme search etc.

An English-Bulgarian dictionary database (a scan of a paper dictionary with about 51000 entries) was used in Smarty because it was freely available and suited for the purpose of this prototype.

Some of the entries include examples of usage, multiword expressions and phrasal verbs, which are parsed and used as resources for multiword expression recognition.

*2) WordNet*

WordNet is a large lexical database, consisting of synonym sets of words – "synsets" –s tructured by part-of-speech and numerous types of semantic relations. The richness of its structural information makes it a highly acceptable resource for various NLP tasks (Mitkov, 2003). In the proposed framework, it provides glosses, which are used as semantic database for word sense disambiguation (WSD).

Semantic relations included in WordNet – hyperonymy, meronymy, synonymy etc. – could be used in future versions to improve the precision of the WSD.

*3) EuroWordNet, BalkaNet etc.*

EuroWordNet is a multilingual set of semantic databases for European languages, which are aligned to WordNet and to each other. It consists of databases for Dutch, Italian, Spanish, French, German, Czech and Estonian.

BalkaNet is a similar set, including Bulgarian, Greek, Romanian, Serbian and Turkish lexical databases.

The links between the lexical databases enable direct translation of specific senses. It also allows multilingual translation within a single framework.

In the implementation discussed in this paper, a small version of Bulgarian BalkaNet is used, consisting of about 15000 synsets. However, the system could easily be extended with other databases from the BalkaNet or EuroWordNet frameworks, thus making it possible for "Smarty" to operate as an English-Greek, English-Romanian, English-Serbian etc. comprehension assistant.

## D. Natural Language Processing Stages

### 1) POS-tagging

Selecting a word in context, instead of copy-pasting or typing in a text box allows POS-tagging to be performed. For languages which exhibit typical ambiguity of lexical categories such as English, this could narrow the set of returned dictionary entries by two or three times.

The POS-tagger in Smarty prototype is SharpNLP – an LGPL .NET library, which was chosen because the system is coded in C#.

### 2) Lemmatisation and Normalisation

This stage saves the user the trimming of words copied from texts and thus speeds up the look-up. Lemmatisation and normalisation are also used in the multiword expression recognition and word-sense disambiguation stages to allow capturing variations.

### 3) Multiword expressions recognition

At this stage the context of the selected word is checked for matches with multiword expressions in the conventional dictionary database. The words from the context are lemmatised and then fuzzy-matched to patterns from a multiword expressions database. Different techniques are applied to compute the degree of match: bag of words, POS-matching, regular expressions. The fuzzy matching algorithm applied imply high recall, still delivering only one or few most relevant multiword expressions, which fit in a tooltip or a small text box.

Automatic multiword expressions recognition capabilities of Smarty allow faster look-up of multiword expressions, compared to the operation of conventional dictionaries, which usually lack such functionalities or capture only exact matches. A sorted list of multiword expressions and phrasal verbs, presented in Smarty, also helps users quickly find wanted translations of multiword expressions.

Using conventional dictionaries, finding out that there is a multiword expression in certain contexts may require the user to scroll and scan the whole dictionary entry by sight. It must be pointed out, that in cases of words with short entries Smarty does not have a significant advantage because a visual scan of possible expressions could also be done in few seconds. However, the advantage of multiword expressions recognition is significant when quering entries of common words like "run", "take", "go", "have" etc., which have many tenths of examples of usage.

### 4) Word-Sense Disambiguation

The ultimate goal of comprehension assistants is to find the most appropriate translation in a given context. Word sense disambiguation contributes to the further narrowing down of the list of possible senses. Figure 3 illustrates how the selected word initially featuring 80 potential meanings has the number of its possible translations reduced to 21 after POS tagging and even further reduced to 1 single possible meaning after correct word sense disambiguation.

In this implementation Smarty uses glosses from WordNet to perform simple WSD in English, related to the method of Lesk (Lesk, 1986) . The framework benefits from the alignment between the lexical databases of WordNet and BalkaNet, which allows word-sense disambiguated sense in English to be mapped directly to precise sense in Bulgarian or other language from BalkaNet or EuroWordNet.

The method for WSD in the prototype of Smarty applies the following algorithm:

1. A word in a text is pointed and then its context is tokenized, normalised and POS-tagged. It is then cleaned from stop-words which are considered to be confusing for the process of WSD.
2. WordNet synonym sets corresponding to the queried word are found in the database and their glosses are extracted.
3. Each gloss is tokenised, part-of-speech tagged, lemmatised/normalised and cleaned from stop-words.
4. The normalised context and the gloss are matched and word-matches are counted.
5. Until there are more glosses, go back to step 3. Otherwise:
6. The gloss with highest number of matches is suggested as the most probable sense. If there are not any matches, the most frequent sense referring to WordNet is suggested. If there are more than one senses with the same number of matches, the most frequent sense is suggested also, again referring to the order of senses in WordNet.
7. The index of the synonym set of the suggested sense is matched to the indices of BalkaNet.
8. If BalkaNet contains the disambiguated sense, then disambiguated translation in Bulgarian is displayed with confidence. Otherwise, other available senses are displayed with a sign of uncertainty.

This algorithm has low precision, due to its simplicity. Disambiguation in English is correct in two cases. The first case is when the context of the queried word includes specific discriminative words from the gloss of the correct sense. The second case is when discriminative words are not present in the context, but the most frequent sense is used, because the most frequent sense is suggested in case of uncertainty.

Precision in WSD to Bulgarian is lower than the precision in English, due to the limited size of the lexical database used–15000 synsets [Windows U1] versus 115000. Also, in most cases BalkaNet includes only one or few most frequent senses for a given word.

Examples of correctly disambiguated senses follow. The words from the glosses which are used to discriminate the sense are underlined.

*- What <u>instrument</u> do you play, Paul?*

*- I play the **bass.***

**Suggested sense*:*** bass – n. the member with the lowest range of a family of musical instruments.

– You are fired! – said the boss.

**Suggested sense:** fire – v. terminate the employment of; "The boss fired his secretary today"; "The company terminated 25% of its workers".

## IV. User Evaluation

Some aspects of Smarty's performance and features were evaluated in real environment by users and compared with two other electronic dictionaries - *SA Dictionary* and *Babylon*.

*SA Dictionary* is a popular English-Bulgarian conventional dictionary, based on the standard simple framework. *Babylon* is an advanced multilingual conventional dictionary framework with graphical user interface having certain similarities to the interface of comprehension assistants – the system captures words clicked anywhere on the screen. Babylon can recognise[Windows U1] phrasal verbs and multiword expressions, but only if they are in the exact form as they appear in the dictionary database. Also, the dictionary lacks NLP functionality for reducing the number of possible translations of single words.

Babylon offers machine translation, however it employs third-party on-line services (probably Systran) and this specific functionalities are not relevant for this evaluation.

### *A. Query time for single words translation*

Smarty framework provides three main quick results in tooltips:

1. The most relevant part-of-speech portion of the entry from a conventional dictionary.

2. A suggested multiword expression which matches the context of the pointed word.

3. Suggested word-sense disambiguated sense in Bulgarian.

A bubble with either a translation from these types appears virtually immediately on the test PC with 1.8 GHz Athlon CPU. The query time is between 0.2 sec to 1.2 sec. This is where the worst cases are met when querying words with the longest list of multiword expressions, due to the time needed to match them to the context.

SA Dictionary also provides virtually immediate results for single-word queries, while Babylon is delayed by a few seconds due to access to Internet resources. However, both lack the capability to reduce the entries to the most relevant sections. This often slows down the time for actual translation as the user is forced to scan long entries with the bare eye.

A small test was conducted in order to assess the time saved with Smarty (if any) in a real environment. Several chapters from Dan Brown's *The Da Vinci Code* were selected, in order to represent a common text with similar style and language complexity. Two native-Bulgarian speakers with different English proficiency read three chapters with Smarty, SA Dictionary and Babylon. Users queried unknown or ambiguous words, and searched their meanings in the entry displayed in dictionary's window (SA Dictionary and Babylon) or in the tooltip, provided by Smarty. The number of queries, the total time needed for the look-up in seconds and the average time per query were computed.

Table I .
User 1 - Undergraduate student

| Chapter | Dictionary | Words | Queries | Time | **Average** |
|---|---|---|---|---|---|
| 2 | SA | 909 | 31 | 153 | **4.94 s/q** |
| 45 | Babylon | 1149 | 23 | 173 | **7.52 s/q** |
| 100 | Smarty | 1183 | 24 | 80 | **3.33 s/q** |

Table II .
User 2 – PhD student

| Chapter | Dictionary | Words | Queries | Time | **Average** |
|---|---|---|---|---|---|
| 2 | SA | 909 | 29 | 74 | **2.55 s/q** |
| 45 | Babylon | 1149 | 28 | 100 | **3.57 s/q** |
| 100 | Smarty | 1183 | 37 | 97 | **2.62 s/q** |

The tables show that the PhD student is much faster than the undergraduate student with all three dictionaries. The results also suggest, that in the experiments carried out by the PhD student,Smarty and SA Dictionary are practically equal in performance, while the undergraduate student translates significantly faster with Smarty. Generally both observations could be explained by the higher English proficiency of the PhD student. The equal speed of operation using Smarty and a conventional] dictionary in one of the tests could be explained by the genre of the texts and by the high English proficiency of the PhD student. Her queries consist of rare words, which are not ambiguous , thus the entries in the conventional dictionary could also be scanned in a moment.

We conjecture that Smarty would perform much faster than conventional dictionaries if tested by language learners with much lower English proficiency. Language learners are expected to query more frequent words, which exhibit higher lexical and part-of-speech ambiguity. This is where Smarty's NLP preprocessing can offer significant advantage over the simple operation of conventional dictionaries, and where Smarty would be most useful.

## Acknowledgment

We would like to express our gratitude to Constantin Orasan, Irina Temnikova and Kiril Kolev for their suggestions and taking part in the evaluation experiments.

## References

[1] H. Feldweg, E. Breidt, "COMPASS An Intel-ligent Dictionary System for Reading Text in a Foreign Language". In F. Kiefer and G. Kiss, editors, Papers in Computational Lexicography, COMPLEX '96, Budapest, pages 53-62.

[2] D. Kolev, "Computer assistant for translators" - Bachelor Thesis (In Bulgarian). Plovdiv University 2005.

[3] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone." In Proc. of the 1986 SIGDOC Conference, pages 24–6, Ontario, Canada.

[4] G. A. Miller, "WordNet: a lexical database for English", Communication of the ACM, Volume 38, Issue 11 (November 1995)

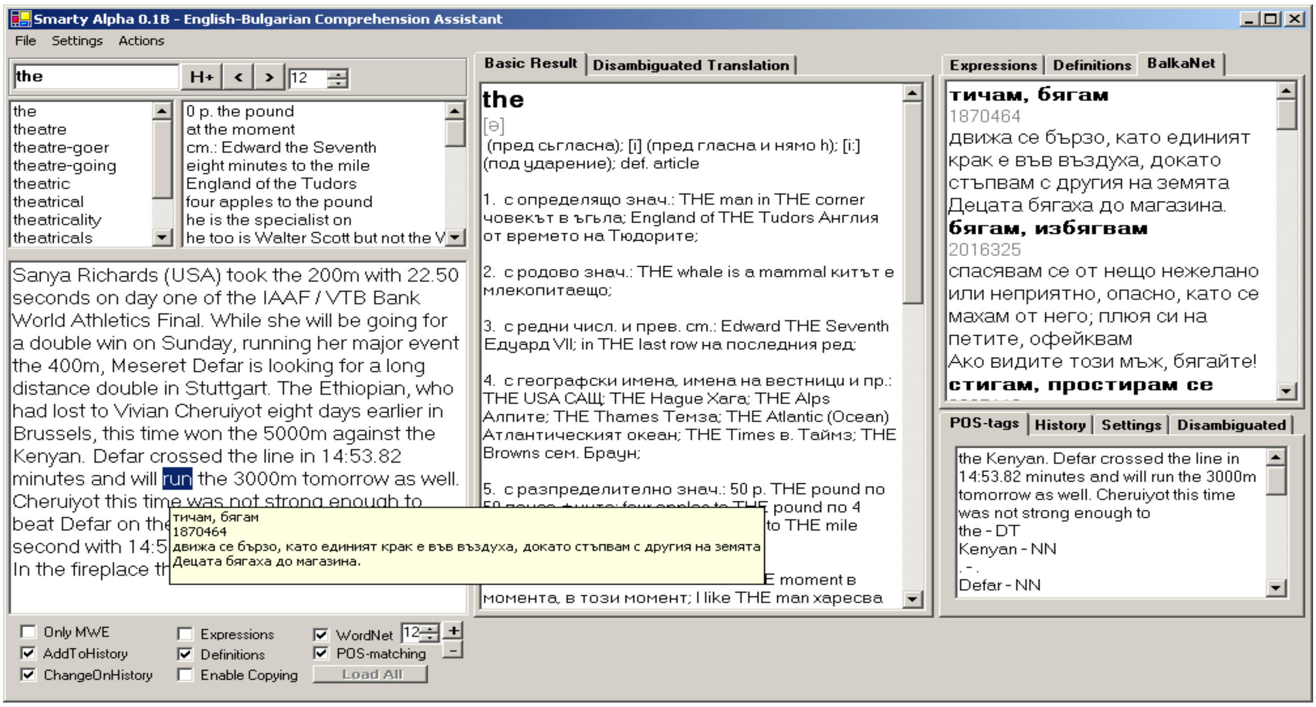
[5] R. Mitkov, R., "The Oxford Handbook of Computa-tional Linguistics", Oxford University Press, 2003.

[6] K. Oflazer, S. Stamou, D. Christodoulakis, “BALKANET: A Multilingual Semantic Network for the Balkan Languages“ – in the Elsnet Newsletter of the European Network in Human Language Technology, 2001.

[7] G. Prószéky, “Comprehension Assistance Meets Machine Translation”. In: Toma3 Erjavec; Jerneja Gros (eds) Language Technologies, 1–5. Institut Jo3ef Stefan, Ljubljana, Slovenia.

[8] P. Vossen, “EuroWordNet: a multilingual data-base with lexical semantic networks”, Kluwer Aca-demic Publishers Norwell, MA, USA, 1998.

## Appendix



Smarty Alpha 0.1B - English-Bulgarian Comprehension Assistant

the

the
theatre
theatre-goer
theatre-going
theatric
theatrical
theatricality
theatricals

0 p. the pound
at the moment
cm.: Edward the Seventh
eight minutes to the mile
England of the Tudors
four apples to the pound
he is the specialist on
he too is Walter Scott but not the V

Sanya Richards (USA) took the 200m with 22.50 seconds on day one of the IAAF / VTB Bank World Athletics Final. While she will be going for a double win on Sunday, running her major event the 400m, Meseret Defar is looking for a long distance double in Stuttgart. The Ethiopian, who had lost to Vivian Cheruiyot eight days earlier in Brussels, this time won the 5000m against the Kenyan. Defar crossed the line in 14:53.82 minutes and will run the 3000m tomorrow as well. Cheruiyot this time was not strong enough to beat Defar on the second with 14:5 In the fireplace th

тичам, бягам
1870464
движа се бързо, като единият крак е във въздуха, докато стъпвам с другия на земята
Децата бягаха до магазина.

**the**
[ə]
(пред съгласна); [i] (пред гласна и нямо h); [i:] (под ударение); def. article

1. с определящо знач.: THE man in THE corner човекът в ъгъла; England of THE Tudors Англия от времето на Тюдорите;

2. с родово знач.: THE whale is a mammal китът е млекопитаещо;

3. с редни числ. и прев. cm.: Edward THE Seventh Едуард VII; in THE last row на последния ред;

4. с географски имена, имена на вестници и пр.: THE USA САЩ; THE Hague Хага; THE Alps Алпите; THE Thames Темза; THE Atlantic (Ocean) Атлантическият океан; THE Times в. Таймз; THE Browns сем. Браун;

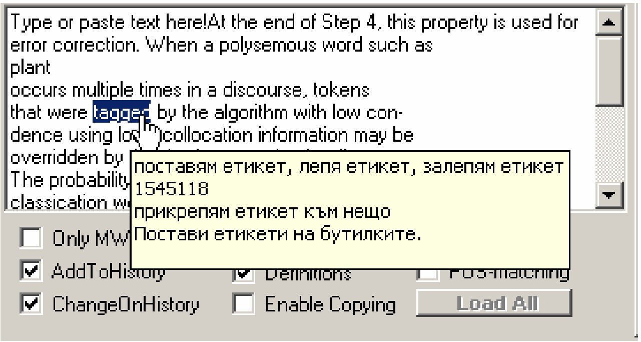
5. с разпределително знач.: 50 p. THE pound по 50 ... pound по 4 ... to THE mile ... E moment в момента, в този момент; I like THE man харесва

**тичам, бягам**
1870464
движа се бързо, като единият крак е във въздуха, докато стъпвам с другия на земята
Децата бягаха до магазина.
**бягам, избягвам**
2016325
спасявам се от нещо нежелано или неприятно, опасно, като се махам от него; плюя си на петите, офейквам
Ако видите този мъж, бягайте!
**стигам, простирам се**

the Kenyan. Defar crossed the line in 14:53.82 minutes and will run the 3000m tomorrow as well. Cheruiyot this time was not strong enough to
the - DT
Kenyan - NN
. - .
Defar - NN

☐ Only MWE
☑ AddToHistory
☑ ChangeOnHistory
☐ Expressions
☑ Definitions
☐ Enable Copying
☑ WordNet 12
☑ POS-matching

Fig 1 . Smarty

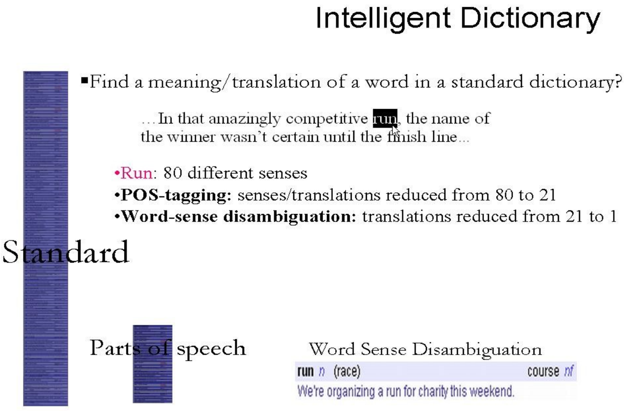


Type or paste text here!At the end of Step 4, this property is used for error correction. When a polysemous word such as
plant
occurs multiple times in a discourse, tokens
that were tagged by the algorithm with low con-
dence using lo collocation information may be
overridden by
The probability
classication w

поставям етикет, лепя етикет, залепям етикет
1545118
прикрепям етикет към нещо
Постави етикети на бутилките.

☐ Only MW
☑ AddToHistory
☑ ChangeOnHistory
☑ Definitions
☐ POS-matching
☐ Enable Copying

Fig 2 . Bilingual word-sense disambiguation



Intelligent Dictionary

- Find a meaning/translation of a word in a standard dictionary?

…In that amazingly competitive run, the name of the winner wasn't certain until the finish line…

- Run: 80 different senses
- **POS-tagging:** senses/translations reduced from 80 to 21
- **Word-sense disambiguation:** translations reduced from 21 to 1

Standard
Parts of speech
Word Sense Disambiguation
run n (race)
course nf
We're organizing a run for charity this weekend.

Fig 3 . The process of narrowing the list of possible translations

ISBN 978-83-60810-14-9
ISSN 1896-7094

# SuperMatrix: a General Tool for Lexical Semantic Knowledge Acquisition

Bartosz Broda, Maciej Piasecki
Institute of Applied Informatics, Wrocław University of Technology, Poland
Email: {bartosz.broda, maciej.piasecki}@pwr.wroc.pl

***Abstract*—The paper presents the SuperMatrix system, which was designed as a general tool supporting automatic acquisition of lexical semantic relations from corpora. The construction of the system is discussed, but also examples of different applications showing the potential of SuperMatrix are given. The core of the system is construction of co-incidence matrices from corpora written in any natural language as the system works on UTF-8 encoding and possesses modular construction. SuperMatrix follows the general scheme of distributional methods. Many different matrix transformations and similarity computation methods were implemented in the system. As a result the majority of existing Measures of Semantic Relatedness were re-implemented in the system. The system supports also evaluation of the extracted measures by the tests originating from the idea of the WordNet Based Synonymy Test. In the case of Polish, SuperMatrix includes the implementation of the language of lexico-syntactic constraints delivering means for a kind of shallow syntactic processing. SuperMatrix processes also multiword expressions as lexical units being described and elements of the description. Processing can be distributed as a number of matrix operations were implemented. The system serves huge matrices.**

## I. Introduction

IF A *WORDNET*[1] for some language does not exists, then . . . it should be created as quickly as possible. This point of view is probably shared by the majority of researchers working in the area of Natural Language Processing. The stand of developers and companies is much less clear, but even them would love to have an occasion to criticise an existing wordnet for not solving all the large scale problems. There are two weakest points of the wordnet in general: its construction is very laborious process, in which skilled lexicographers must be involved, and it takes a lot of time to construct a new wordnet, even if we start with translating a wordnet built for another language (i.e. mostly the English wordnet). Both problems are strictly correlated. While starting a project on the construction of the Polish Wordnet [2], called plWordNet (or *Słowosieć* in Polish), we decided to build it from scratch, in order to construct it as a faithful description of the Polish lexical semantic relations. So we increased the amount of work to be done, but in the same time we did not increase the amount of money assigned to the project (anyway, as it was quite moderate, so it was not a big difference). However, from the very beginning we planned to support the work of lexicographers by different types of language tools automatically constructed on the basis of large corpora:

- delivering some means of intelligent semantic browsing across *lexical units*[2] (henceforth LU),
- or even suggesting to the lexicographer some lexical semantic relations between LUs or groups of LUs (e.g. wordnet synsets).

Browsing on the basis of LU meaning relations requires some way of measuring *semantic relatedness* between pairs of LUs. Following Edmonds and Hirst [3] we prefer the term semantic relatedness instead of the widely used term of *semantic similarity*, because the former better expresses the nature of a numerical measure one extracts from corpora. A Measure of Semantic Relatedness (henceforth MSR) is a function which for a given pair of LUs returns some real number expressing how semantically close the elements of the given pair are, regardless of the exact nature or cause of this relation:

$$MSR : L \times L \to R \tag{1}$$

where $L$ is the set of lexical units and $R$ is the set of real numbers.

As our objective is to build a set of language tools supporting wordnet construction, we will limit the rest of our considerations only to the automatic extraction of instances of lexical semantic relations from corpora. There are two main paradigms of automatic extraction of instances of lexical semantic relations, e.g. [4]:

- *pattern-based*,
- and *clustering-based*, called also distributional paradigm, as it originates directly from the *Distributional Hypothesis* formulated by Harris [5].

According to the pattern based approaches there are some lexico-syntactic patterns, which combine two LUs and mark the two LU as an instance of some lexical semantic relation, e.g. hypernymy, see e.g. a seminal work of Hearst [6]. So only one occurrence of a precise pattern can signal the given association of LUs.

The clustering-based approaches assume that the similarity of distributions of some LUs across different lexico-syntactic or even semantic contexts is evidence for their close semantic relation. The stronger the similarity is the closer the LUs are

[1] By wordnet we mean here an electronic thesaurus of a structure following the main lines of the Princeton WordNet thesaurus [1].

[2] A lexical unit is a one word or multiword lexeme named in the lexicon by its morphological base form and representing a whole set of one word or multiword forms possessing the same meaning and differing in the values of morphological categories.

in their meaning. The name of the paradigm emphasises that we are looking for similar distributions of LUs and, in some way, we cluster them into groups of highly semantically related LUs.

There are plenty of methods proposed for the automatic extraction of Measures of Semantic Relatedness. All start with processing a corpus and constructing a coincidence matrix describing co-occurrences of LUs (rows) and lexico-syntactic contexts (columns). They differ in three aspects: definitions of contexts, transformations of the raw frequencies and calculation of the final measure value. At the beginning of our project project it was completely unclear which known MSRs perform better, and which of them would work for Polish. Polish is not only a language using alphabet extended in comparison to the ASCII code[3], is a language typologically different than English, but also is a language with fewer language tools and resources than English. The third problem was the worst, as the difference between English and Polish in this area is huge. Thus we decided to construct a system capable of utilising existing Polish language tools, Polish corpora and reimplementing, investigating and evaluating as many MSRs as possible.

The goal of the paper is to present the constructed system called *SuperMatrix* and discuss its various successful applications. As we would like to make SuperMatrix free for research uses, we hope that the latter can guide potential users to the areas of its applications. The first experiments done with the help of SuperMatrix were presented in [7], and the general scheme of processing was discussed in [8]. Here we are going to present the first thorough description of SuperMatrix.

## II. Blueprint for the Construction of MRSs

There is a plethora of approaches to extraction of similarity between LUs, e.g. [9]–[12]. Basically, they all follow similar pattern for construction. This general blueprint as implemented in SuperMatrix is shown on Figure 1. Following the idea of *distributional similarity* and Distributional Hypothesis [5], first, co-occurrence data is collected from text corpora for selected words.

There are three main approaches to represent a context. One can count words occurring together with the given LU inside a passage of text, e.g. inside a paragraph or document. This approach has been used in the technique called *Latent Semantic Analysis (LSA)* [13]. Another popular method for context representation is counting words co-occurring inside a text window, this approach is called a *Word Space* [14]. In *Hyperspace Analogue to Language (HAL)* [15] smaller weights are assigned to words if they occur further from the centre of the context — the centre is occupied by the LU being described.

Very good results were observed after enriching description of a context with syntactic information, e.g. [8], [9], [12]. This method counts only co-occurrences between an LU in the context centre and selected lexico-syntactic relations in which the LU is involved.

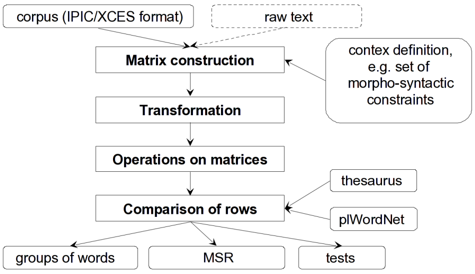


Fig. 1. General blueprint for creation of measures of semantic relatedness.

Usually LUs are represented as feature vectors in a high dimensional space. Each feature corresponds to a single context, in which an LU occurs in corpus. It is convenient for further processing to think about the collection of feature vectors in terms of a matrix $M$ (see figure 2), consisting of $i$ rows (words) described by $j$ features (context). The value of $M[n_i, c_j]$ tells how many times the word $n_i$ occurred in the context $c_j$.

In the next step co-occurrence matrices are used to calculate similarity between words. There are many methods for doing this. One can measure the Euclidean distance between word vectors, calculate cosine between vectors to measure how close they are one to another, etc. [16].

Experiments showed that raw frequency counts of co-occurrences are not very useful from the perspective of lexical semantic knowledge acquisition. First, after the analysis of the collected data it can become apparent that not all the features used are descriptive enough to differentiate between LU meanings. That is why an additional step of *filtering* features is often preformed [8], [12]. Second, there is a need to emphasise an inner structure (or a latent structure) of the data before comparing word vectors. There are two main approaches to this problem: based on *transformation* and *weighting*.

One of the well known example of transformation is *Singular Value Decomposition* [17]. It is a method for reducing matrix dimensionality, and was applied in LSA to achieve a form of generalisation from the raw frequency counts.

When comparing nouns one can quickly arrive to the conclusion, that almost every noun can be modified by "*liczny*" (*numerous*, *countless*), but "*bezołowiowy*" (*unleaded*) will be a feature of a few very specific LUs. This is where the weighting is helpful. The basic idea of weighting is to assign greater weights to features that are more descriptive than the others.

The above steps are required to create an MSR. But rarely the construction of MSR is the main aim of one's work. Usually it is only a means for achieving some other goal. For example one can cluster LUs in order to semi-automatically extend lexicons [18] or use similarity for the correction of medical handwritten documents [19].

Last, but not least, there remains a question about comparing different MSRs. There are three main approaches to evaluation of MSRs [20], [21]: mathematical analysis of

[3] In 2005, when we started, many similar systems did not process the extended ASCII code, not mentioning UTF-8.

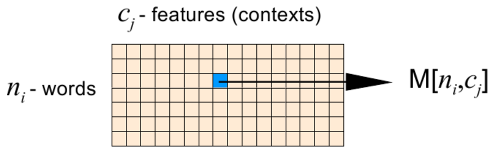


Fig. 2. Schematic view of co-occurrence matrix.

their formal properties, application specific evaluation and comparison with human judgement. For example, Landauer and Dumais used the third approach for the evaluation of LSA. They used a synonymy part of the *Test of English as a Foreign Language (TOEFL)* to test the ability to automatically differentiate between synonymous and non-synonymous LUs. Because TOEFL is limited in the number of questions it includes and is available only for English, a *WordNet Based Synonymy Test (WBST)* was proposed to generate “a large set of questions identical in format to those in the TOEFL” [22]. An analysis of WBST-based approach to evaluation of MSRs is presented in [21].

An instance of the WBST test is built thus: first, a pair of LUs: $\langle q, s\rangle$, is chosen from a wordnet (*WordNet* 2.0 in [22]), where $q$ is a question word, and $s$ is a randomly chosen synonym of $q$; next, three other words that are not in synsets of $q$ and $s$ are randomly drawn from the wordnet — they are a detractor set $D$. The task for MSR is to point which word from the set $A = D \cup \{s\}$ is a synonym to $q$. For example for the word *administracja (administration)* $A$ consists of: *poddasze* (attic), *repatriacja* (repatriation), **zarząd** (board, management) and *zwolennik* (follower, zealot).

## III. SuperMatrix

SuperMatrix is a collective name for a set of libraries for programmers and end-user tools for creation, storing and manipulation of co-occurrence matrices describing distributional patterns of LUs.

Overall, the implementation and design has been dictated by requirements placed upon SuperMatrix. Above all the system should be *extensible* and *flexible*. This property is necessary for experimenting with different methods of MSR extraction. *Efficient* processing is also crucial, as statistical methods tends to yield better results with the increasing amount of data [23]. Thus, the software has been written in C++, with additions of Python bindings and helper scripts.

Because we are working in heterogeneous environment we wanted the system to be as much *portable* as possible. This could also lead to lessen effort in embedding parts of the system in end-user applications. After conducting a few experiments we realized that ability to quickly test new algorithms would be very convenient, so we added *fast prototyping* to the requirement list.

SuperMatrix consists of several modules, namely:

- Matrices – a library for storing matrices.
- Comparator – a library enabling computation of similarity between rows of matrices (i.e. between LUs) using different MSRs.
- Set of tools, including (but not limited to) tools for creation of matrices: LUs by features, tools for evaluating of MSRs, tools for joining different matrices and analysis of the matrix content, e.g. manually browsing selected rows and columns from any matrix, including transformed and weighted matrices.
- Clustering – package consisting of several clustering algorithms. SuperMatrix can interact with CLUTO [24] and perform Clustering by Committee(CBC) [18], RObust Clustering using linKs (ROCK) [25] and Growing Hierarchical Self-Organising Maps (GHSOM) [26]. We reimplemented ROCK and GHSOM with little modification, CBC was reimplemented as well as significantly extended [27].
- Set of helper scripts and SWIG[4] wrappers for main classes of the Matrices and Comparator libraries.

For portability reasons we wanted to avoid external dependencies as much as possible. Only required components are open-source and cross-platform. CMake[5] is used for our build system. SuperMatrix is also heavily dependant on availability of Boost libraries[6], Other used software packages are not so crucial for SuperMatrix, e.g. SWIG is needed only for generating Python wrappers, CLUTO for clustering. For the construction of matrices from Polish corpora we also use parts of TaKIPI [28] engine – an open-source mopho-syntactic tagger for Polish[7]. We have tested SuperMatrix under different flavours of Linux as well as Microsoft Windows.

This system has been under active development for almost two and a half years now. At the time of writing it consists of almost 24 thousands lines of C++ code and almost 3.5 thousands of lines of code written in Python.

### A. Matrices

Most fundamental question for software toolkit performing heavy computation on matrices is: how to represent a matrix object in computer memory? There are many options for doing this, i.e. one can store it in a dense or sparse format, for a sparse format there exists many possible representations. Not wanting to be bound to only one implementation we have defined set of operations that a matrix has to be able to perform and tested several different approaches.

A dense format was used for small matrices. *Compressed Column Storage* (CCS) [17] is not very convenient for a matrix whose content is being changed constantly. During first experiments [7] we created implementation for storing matrix in database. We tested three possible representations (storing columns, rows or non-zero cells of a matrix), but performance overhead was too large for practical usage. So we have removed support for the storage in database.

It appeared that the most powerful (in terms of flexibility and efficiency) representation is the one using `map` collection of the standard C++ library. Thanks to guaranteed

[4] Simplified Wrapper and Interface Generator, http://www.swig.org/
[5] http://www.cmake.org/
[6] http://www.boost.org/
[7] Available for download at http://plwordnet.pwr.wroc.pl/g419/tagger/

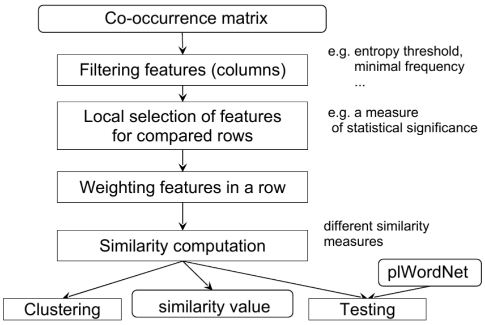

Co-occurrence matrix
Filtering features (columns)
e.g. entropy threshold, minimal frequency ...
Local selection of features for compared rows
e.g. a measure of statistical significance
Weighting features in a row
Similarity computation
different similarity measures
plWordNet
Clustering
similarity value
Testing

Fig. 3. Framework for computation of MSR.

$O(log(n))$ complexity we have achieved both flexible and efficient implementation of a matrix. We have encapsulated functionality of a word feature vector using this representation in `CompressedVector` class. Matrices using this representation were called `CCSMatrix` (for storing matrix columnwise) and `CRSMatrix` (for storing in row fashion).

To improve performance we cache some information together with vectors of features. Most importantly we keep entropy of a vector and sum of vector cells. A `SuperMatrix` class is a composition of such cached data (`Feature` class) and `CRSMatrix` (`CCSMatrix`).

Classes in the namespace `Matrices:IO` are capable of saving matrices in a few popular format. Native SuperMatrix format is based on sparse format used by CLUTO [24]. This makes interaction with CLUTO easier. Matrix can also be exported to CCS or CRS format, which enable interaction with such tools like Infomap NLP [29], SenseClustres [11] or SVDPACK [17].

In the current representation we have been able to perform efficient computation[8] using matrices for 13 thousands words described by more then 270 thousands of features created on the basis of a corpus consisting of more then 550 million words[9].

### B. Comparator

Comparator library is used for calculating similarity values between rows of a matrix. It is extensible library used for constructing and testing different approaches to computation of MSRs.

A typical framework for processing is presented on Figure 3. After creation of a matrix some global filtering of columns is performed. It follows intuition that some features are not good discriminators for a matrix row. SuperMatrix can do filtering using three methods:

- Using a stoplist. This step is performed by most of the algorithms, even if it is not implicitly mentioned. Stoplist consists of functional words (like conjunction or prepositions) and words that are not good meaning bearers. Usually this step is performed before creation of a matrix, but we allow to do it afterwords.
- Filtering using the *minimal global term frequency* of a word. Most statistically based methods do not cope well with events of extremely low frequencies, so those events are usually treated as outliers and are removed from dataset.
- Filtering using entropy of a column as a measure of noise it introduces. We used Shannon's entropy:

$$Entropy_w = -\sum_i p_{w,i} \cdot \log p_{w,i}, \qquad (2)$$

  where $p_{w,i}$ is probability of occurrence of the $i$'th feature with the word $w$. Entropy is maximised for events of maximal uncertainty — here for features that do not differentiate good between rows of a matrix.

We observed that, some features are globally good discriminators, but for certain LUs they are caused by some accidental frequencies (e.g. an error of morpho-syntactic disambiguation, sentence boundary detection or a simple spelling mistake). That is why we have isolated yet another step in the process, namely *local feature selection*. It is implicitly present in co-occurrence retrieval models [12] (CRM). The feature sequences selected from the rows for both LUs may have to be padded (usually with zeroes), if the similarity measure requires equal-size vectors.

Z-Score (variant of t-score) can be used as a measure of association between an LU and a feature. If the observed frequency of occurrence of some feature with some LU is significantly greater then expected, then this feature is a good discriminator for this LU.

In the following step we performed weighting of feature vectors of LUs. This stage aim for emphasising important data in matrix. Several weighting schemes were implemented in SuperMatrix:

- Term frequency – inversed document frequency is a popular method for decreasing weights of the very frequent words used in Information Retrieval (IR) for assigning lower scores to words occurring in all documents:

$$tf.idf_{w,d} = tf_{t,d} \cdot \log \frac{N}{df_t}, \qquad (3)$$

  where $tf_{t,d}$ is a number of occurrences of the word $t$ in the document $d$, $df_t$ is the number of documents containing the word $t$, $N$ is number of documents. This weighting scheme has been used mainly for processing of document by words matrices.
- In LSA [13] before reduction of the dimension, a matrix has been weighted in two-step process. First the cells of a matrix were scaled logarithmically: for each $i$, $j$: $M[w_i, c_j] = \ln(M[w_i, c_j] + 1)$ and divided by entropy of a row of a matrix. We use *logent* as a name for this weighting scheme.
- Z-Score (or t-score) can be used not only for local selection of features, but also as a weighting function, e.g. [8], [33].

[8] On a contemporary PC, i.e. 1,5 GHz processor with 1 GB of RAM.

[9] 254 millions words from IPI Pan Corpus [30], 100 millions from Rzeczpospolita [31] (Polish newspaper), and 100 from the Polish edition of Wikipedia [32]

- Another popular family of weighting schemes is based on *Mutual Information* (MI). In [9] some formal introduction for MI in the context of extraction of MSRs is presented. In *The Sketches Engine* [34] this measure was used for the generation of a distributional thesauri. In [18] a variant of this measure called *Pointwise Mutual Information* was used (extended with a *discounting factor*).
- Some measures of similarity operate in probability space. There are few methods for transition from frequencies to probabilities. We used for this purpose *Maximum Likelihood Estimation*.
- To reimplement best faring MSRs from [12] we added weighting schemes based on $CRM_{MI}$ and $CRM_{dt}$.
- During experiments with different MSRs we noticed [8] that feature values in the matrix depend too directly on frequencies. However no corpus is perfectly balanced, and any weighting function alone does not solve the problem. We need some generalisation from the raw frequencies. Applying SVD to very sparse matrices does not help [7]. We assumed that similarity of two types of objects depends more on which significant features characterise them than on the exact numerical values of those features' "strength". So, we developed *Rank Weight Function*(RWF) – a weighting scheme that builds relative ranking of importance of features from raw frequencies[10]. Experiments showed that for Polish the MSR based on RWF produces better results [35], in the WBST test. SuperMatrix supports methods enabling transition into the rank space.

After weighting of a matrix one can perform similarity computation. We implemented several similarity functions, e.g. the commonly used, geometry inspired *cosine function*, $SIM_{IRAD}$ – function using divergence of the two probability distributions, $SIM_{CRM}$ for the reimplementation of the co-occurrence retrieval models (CRM), Lin's measure based on information theory, etc. Surprisingly, cosine performed very well in comparison to other functions.

On the figure 3 one step is not shown: transformation of a matrix. Transformations are usually computationally intensive, so most of the time they are performed independently of typical process shown described earlier. We use a few methods of transformation in current version of SuperMatrix, namely:

- Singular value decomposition using SVDPACKC [17].
- Transformation of a LUs by features matrix into the similarity matrix. This transformation is useful for example during exporting data into CLUTO for clustering.
- Relative Frequency Focus — a transformation required for reimplementation of the approach presented in [36].

As a final note on Comparison module we want to emphasise that the framework presented on Figure 3 is not fixed in SuperMatrix. Our framework seems to encompass many, if not all, methods of MSR construction. For instance, to reimplement CRM, one needs the identity function for global selection, some weight function analysed in CRM for transformation, local selection by the condition $\mathbf{M}[w_i, c_j] > 0$ (applied after transformation) and the CRM F-score as the similarity measure.

### *C. Tools*

This section selectively describes tools available in SuperMatrix package for usage with little to no coding at all.

*1) Architect:* Collective name for applications used to create different kinds of co-occurrence matrices. Using tools in this category we can create:

- a documents by words matrix (for document clustering or Information Retrieval),
- a window matrix, i.e. a matrix in which context describes co-occurrence of words in text window of fixed sized (e.g. 5 words to the left and 5 words to the right from the target word),
- a HAL-like matrix — a matrix created in a similar manner to window matrix, but higher scores are assigned to words occurring closer in the text window,
- a sentence matrix — a window matrix with non-fixed size of a text window, in which only co-occurrences in the same sentence are counted,
- a matrix describing LUs by co-occurrence of those LUs in syntactic relations.

Because there is no robust parser available for Polish, for the creation of a matrix describing co-occurrences of LUs in syntactic relation we used a simplified approach based on defining morpho-syntactic constraints. Those constraints are expressed in JOSKIPI — a specialised language developed for TaKIPI [28] — a Polish morpho-syntactic tagger.

It is worth noting that Architect can create matrix for LUs as well as for multiword expressions (MWE). We require only a limited description of syntactic dependencies between constituents of MWEs. In SuperMatrix a module for the automatic extraction of those dependencies is available (for the description of this method see [37])

As the amount of textual data is increasing, processing of a raw text (or annotated text in XML format) is becoming performance bottleneck. To speed up the process of matrix construction SuperMatrix can read binary format generated by Poliqarp [38].

*2) WBST Tester:* A tool performing evaluation of MSRs by the application of the *WordNet Based Synonymy Tests (WBST)*. With addition of a few Python scripts we can generate WBST, test MSRs and perform test on statistical difference of the results produced by different MSRs.

*3) Summator:* A tool for joining different matrices. We observed that combining several matrices created with usage of different morpho-syntactic constraints resulted in better MSRs [21].

*4) VectorExtractor:* A tool for supporting manual analysis of a matrix content. i.e. manual browsing of parts (rows and columns) of huge matrices.

*5) Relations:* SuperMatrix includes also a set of tools for preparing training data used during training classifiers processing pairs of LU and assigning them to different types

[10] For detailed description of RWF see [8], [35].

of wordnet relations, see [39]. Features extracted on the basis of raw matrices or transformed matrices are next stored in the ARFF format which is supported by many Machine Learning systems e.g. Weka system [40] used in [39]

*6) Simbuilder:* In many applications it is easier first to transform the matrix describing LUs by row vectors of features into the square matrix of LU similarity, e.g. the generation of the list of the most similar LUs to the given one applied in the initial phase of CBC [18], or calculation of weights in the RFF MSR [36].

This last task has complexity of $O(n^2)$, where $n$ is dependent on the number of LUs in the matrix. For large matrices the expected time of transformation to the domain of similarity is barely acceptable. For solving this problem, one can apply several approaches, e.g. based on some heuristics that are introduced to increase the speed of performed computations, e.g. [33], [34]. Mostly, precision become a little decreased, but the time of processing is reduced a lot.

From the point of view of the extraction of MSR for the needs of the construction of a lexical semantic network, precision is most important, so its decrease cannot be accepted. Fortunately, the calculation of the LU similarity matrix can be easily distributed. We constructed a tool that can divide the whole task into several computers or processing nodes in a cluster of computers. Communication between processes is based on the Message Passing Interface.

## IV. Usage examples

The primary application of SuperMatrix is construction of tools for semi-automatic extraction of instances of lexical semantic relations used next in extending plWordNet [41]. Mostly the system is used for the construction of different MSRs [7], [8], [21], [35], but also was applied to clustering text documents [42].

SuperMatrix was applied to the construction of MSRs for: Polish nouns [7], [8], [21], verbs and adjectives [35]. The system was also used in the development of the Rank Weight Function [8], which was next implemented in it. SuperMatrix was used for preparing training data for the construction of classifiers of types of lexical semantic relations [39], e.g. hypernymy, meronymy etc.

An interesting application was the support for the construction of a corpus on the basis of documents from the web. SuperMatrix was applied to construct a tool discovering duplicates of documents in a semi-automatic way. For the documents downloaded from the web[11] a matrix: documents by words was built. Next the matrix was transformed to the similarity matrix. On the basis of the similarity matrix pairs of documents whose similarity was above some defined threshold were stored in a file sorted in the descending order of their similarity. The high similarity of documents was a precise signal of duplication, so it was enough to manually check some of the top documents in order to remove duplicates.

Finally, SuperMatrix was utilised in the project whose goal was to develop an OCR of handwritten medical documents. A language model based on the distributional semantic similarity of words was built on the basis of SuperMatrix [19]. The model was next used for the correction of recognition done on the graphical level. A sequence of token positions was delivered to the system. Each token position was assigned a list of potential recognitions for this position. The semantic language model built on the basis of the domain corpus was used in the algorithm called SemWnd, which tried to find a sequence of potential recognitions maximising the semantic consistency of the sequence.

## V. Existing systems

There exists a few software solutions that can satisfy some of design requirements stated in Section I. Reimplementation of LSA for Polish [43] was performed using the combination of *MC Toolkit* [44] and SVDPACKC [17]. We have stumbled upon few problems with that combination. Most important, MC Toolkit supported only ASCII encoding and could only create a words by documents matrix. Also SVD approach is computationally expensive, we were able to reduce dimensions of a matrix describing only four thousands nouns appearing in about 180 thousands short documents [7]. *Infomap NLP* package [29] supports similar functionality to MC Toolkit with SVD, but it was especially created for the extraction of word meanings from free text corpora. One noticeable improvement over MC Toolkit is ability to operate in Word Space. It suffered from similar limitations because of using combination of MC Toolkit with SVDPACKC. Additionally we had problems with building and installing this system.

Recently, Infomap NLP package has been abandoned in favour of a new system called *Semantic Vectors* (SV) [10]. Main differences with Infomap NLP are: usage of random projection instead of SVD for dimensionality reduction and the implementation, which was written completely in Java, using Apache Lucene as a document-indexing engine. Although Semantic Vectors looks interesting it currently dose not support many MRSs. Also, it does not support other methods for the description of context than co-occurrences in document or text window. First public release of SV happened in October, 2007, when SuperMatrix had most of its functionality already implemented.

Natural Language Toolkit (NLTK) [45] is a popular software for performing fundamental natural language processing task. It does not fully support statistical lexical semantic knowledge acquisition and does not support any Polish corpora.

SenseClusters (SC) [11] is the most relevant and similar package to SuperMatrix. It is a collection of Perl modules and programs for clustering of similar words (and contexts) based on distributional similarity. It supports couple of MSRs, creation of Word Space matrices and interacts very well with SVDPACKC for dimensionality reduction and CLUTO for clustering. Unfortunately it does not support Unicode, and cannot use morpho-syntactic constraints or a similar mechanism for the definition of features during matrix construction.

[11] First the downloaded documents were filtered according to the presence of too many words not recognised during the morphological analysis

Sketch Engine (SE) [34] provides support for distributional thesauri out of text corpora. At the time of design phase of SuperMatrix SE did not provide full support for multiword expressions. For computation it used a modification of Lin's measure and currently it uses modification of the Dice coefficient for performance reasons. Being a commercial, closed-source application, we are not aware of possibilities of experimenting with self-made MSRs inside it. Keeping data on an external server would be inconvenient for our needs too.

## VI. Conclusions

We have presented SuperMatrix — a general tool for the acquisition of lexical semantic knowledge from text corpora. At the end of the project we would like to release the version 1.0 for public usage under a free research licence. Commercial licenses are already available.

Possible application areas for SuperMatrix include already mentioned extraction of MSRs from corpora and semantic correction of handwritten text. This system can be also used to perform unsupervised word sense disambiguation, named entity disambiguation, sentiment analysis, document indexing, clustering and retrieval and search engine construction. One can even use SuperMatrix for the extraction of lexical semantic relation in a way following the pattern-based paradigm, e.g. lexico-syntactic patterns expressed in JOSKIPI are used as features describing matrix columns.

We suspect that our system can be also used outside the domain of natural language processing, i.e. everywhere were an object can be represented as a feature vector (especially in high dimensional space).

We plan to use and extend SuperMatrix in upcoming research projects. Because of the highly parallel nature of processing we will extend tools in a way enabling flexible computation in distributed environment via Message Passing Interface (MPI).

We also want to create tools for the creation of matrices for other languages. As a primary goal we will focus on English, with possible addition of other languages. Also we will extend Comparator module with additional weighting schemes and similarity functions.

## Acknowledgment

Work financed by the Polish Ministry of Education and Science, project No. 3 T11C 018 29.

## References

[1] C. Fellbaum, Ed., *WordNet — An Electronic Lexical Database*. The MIT Press, 1998.

[2] M. Derwojedowa, M. Piasecki, S. Szpakowicz, and M. Zawisławska, "Polish WordNet on a shoestring," in *Proceedings of Biannual Conference of the Society for Computational Linguistics and Language Technology, Tübingen, April 11âĂ¿13 2007*. Universität Tübingen, 2007, pp. 169–178.

[3] P. Edmonds and G. Hirst, "Near-synonymy and lexical choice," *Computational Linguistics*, no. 28(2), pp. 105–144, 2002. [Online]. Available: http://ftp.cs.toronto.edu/pub/gh/Edmonds+Hirst-2002.pdf

[4] P. Pantel and M. Pennacchiotti, "Espresso: Leveraging generic patterns for automatically harvesting semantic relations." ACL, 2006, pp. 113–120. [Online]. Available: http://www.aclweb.org/anthology/P/P06/P06-1015

[5] Z. S. Harris, *Mathematical Structures of Language*. New York: Interscience Publishers, 1968.

[6] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora." in *Proceeedings of COLING-92*. Nantes, France: The Association for Computer Linguistics, 1992, pp. 539–545.

[7] M. Piasecki and B. Broda, "Semantic similarity measure of Polish nouns based on linguistic features," in *Business Information Systems 10th International Conference, BIS 2007, Poznan, Poland, April 25-27, 2007, Proceedings*, ser. Lecture Notes in Computer Science, W. Abramowicz, Ed., vol. 4439. Springer, 2007.

[8] M. Piasecki, S. Szpakowicz, and B. Broda, "Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns," in *Proc. Text, Speech and Dialog 2007 Conference*, ser. LNAI, vol. 4629. Springer, 2007.

[9] D. Lin, "Automatic retrieval and clustering of similar words," in *COLING 1998*. ACL, 1998, pp. 768–774. [Online]. Available: http://acl.ldc.upenn.edu/P/P98/P98-2127.pdf

[10] D. Widdows and K. Ferraro, "Semantic vectors: a scalable open source package and online technology management application," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, E. L. R. A. (ELRA), Ed., Marrakech, Morocco, may 2008.

[11] A. Purandare and T. Pedersen, "Senseclusters - finding clusters that represent word senses," in *HLT-NAACL 2004: Demonstration Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 – May 7 2004, pp. 26–29.

[12] J. Weeds and D. Weir, "Co-occurrence retrieval: A flexible framework for lexical distributional similarity," *Computational Linguistics*, vol. 31, no. 4, pp. 439–475, 2005.

[13] T. LANDAUER and S. DUMAIS, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological review*, vol. 104, no. 2, pp. 211–240, 1997.

[14] H. Schütze, "Word space," in *Advances in Neural Information Processing Systems 5*, S. Hanson, J. Cowan, and C. Giles, Eds. Morgan Kaufmann Publishers, 1993. [Online]. Available: citeseer.ist.psu.edu/schutze93word.html

[15] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 203–208, 1996.

[16] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 2001.

[17] M. Berry, "Large scale singular value computations." *International Journal of Supercomputer Applications*, vol. 6, no. 1, pp. 13–49, 1992.

[18] P. Pantel, "Clustering by committee," Ph.D. dissertation, Edmonton, Alta., Canada, Canada, 2003, adviser-Dekang Lin.

[19] B. Broda and M. Piasecki, "Correction of Medical Handwriting OCR Based on Semantic Similarity?" *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4881, p. 437, 2007.

[20] T. Zesch and I. Gurevych, "Automatically creating datasets for measures of semantic relatedness," in *Proceedings of the Workshop on Linguistic Distances*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 16–24. [Online]. Available: http://www.aclweb.org/anthology/W/W06/W06-1104

[21] M. Piasecki, S. Szpakowicz, and B. Broda, "Extended similarity test for the evaluation of semantic similarity functions," in *Proceedings of the 3rd Language and Technology Conference, October 5–7, 2007, Poznań, Poland*, Z. Vetulani, Ed. Poznań: Wydawnictwo Poznańskie Sp. z o.o., 2007, pp. 104–108.

[22] D. Freitag, M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, and Z. Wang, "New experiments in distributional representations of synonymy." in *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 25–32.

[23] J. Curran and M. Moens, "Scaling context space," 2002.

[24] G. Karypis, "CLUTO—a clustering toolkit," Tech. Rep. #02-017, Nov 2003.

[25] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," *Information Systems*, vol. 25, no. 5, pp. 345–366, 2000. [Online]. Available: citeseer.ist.psu.edu/guha00rock.html

[26] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing maps: exploratory analysis of high-dimensional data," 2002.

[27] B. Broda, M. Piasecki, and S. Szpakowicz, “Sense-based clustering of polish nouns in extracting semantic relatedness,” June 2008, delivered to the AAIA’08 workshop at IMCSIT conference.

[28] M. Piasecki, “Polish tagger TaKIPI: Rule based construction and optimisation,” *Task Quarterly*, vol. 11, no. 1–2, pp. 151–167, 2007.

[29] D. Widdows, “Unsupervised methods for developing taxonomies by combining syntactic and statistical information,” in *NAACL ’03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 197–204.

[30] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Warsaw: Institute of Computer Science, Polish Academy of Sciences, 2004.

[31] “Korpus rzeczpospolitej,” [on-line] http://www.cs.put.poznan.pl/dweiss/rzeczpospolita.

[32] “Wikipedia,” [on-line] http://pl.wikipedia.org.

[33] J. Curran, “From Distributional to Semantic Similarity,” Ph.D. dissertation, Ph. D. thesis, University of Edinburgh, 2004.

[34] A. Kilgarriff, P. Rychly, P. Smrz, and D. Tugwell, “The Sketch Engine,” *Information Technology*, vol. 105, p. 116, 2004.

[35] B. Broda, M. Derwojedowa, M. Piasecki, and S. Szpakowicz, “Corpus-based semantic relatedness for the construction of polish wordnet,” in *Proceedings of the 6th Language Resources and Evaluation Conference (LREC’08)*, 2008, to appear.

[36] M. Geffet and I. Dagan, “Vector quality and distributional similarity,” in *Proceedings of the 20th international conference on Computational Linguistics, COLING2004*, 2004, pp. 247–254.

[37] M. D. Bartosz Broda and M. Piasecki, “Recognition of structured collocations in an inflective language,” in *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2007.

[38] D. Janus and A. Przepiórkowski, “Poliqarp 1.0: Some technical aspects of a linguistic search engine for large corpora,” *The proceedings of Practical Applications of Linguistic Corpora*, 2005.

[39] M. Piasecki, S. Szpakowicz, M. Marcińczuk, and B. Broda, “Classification-based filtering of semantic relatedness in hypernymy extraction,” in *Proceedings of the GoTAL 2008*, ser. LNCS. Springer, 2008.

[40] Weka, “Weka 3: Data Mining Software in Java,” 2008, http://www.cs.waikato.ac.nz/ml/weka/.

[41] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda, “Words, concepts and relations in the construction of Polish WordNet,” in *Proceedings of the Global WordNet Conference, Seged, Hungary January 22–25 2008*, A. Tanâcs, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen, Eds. University of Szeged, 2008, pp. 162–177.

[42] B. Broda and M. Piasecki, “Experiments in documents clustering for the automatic acquisition of lexical semantic networks for polish,” in *Proceedings of the 16th International Conference Intelligent Information Systems*, 2008, to appear.

[43] M. Piasecki, “LSA based extraction of semantic similarity for Polish,” A. Zgrzywa, Ed. Oficyna Wydawnicza Politechniki Wrocławskiej, 2006, pp. 99–107.

[44] I. S. Dhillon and D. S. Modha, “Concept decompositions for large sparse text data using clustering,” *Machine Learning*, vol. 42, no. 1, pp. 143–175, Jan 2001.

[45] S. Bird, “NLTK: The Natural Language Toolkit,” in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 69–72.

[46] A. Zgrzywa, Ed., *Proceedings of Multimedia and Network Information Systems*. Oficyna Wydawnicza Politechniki Wrocławskiej, 2006.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Definition Extraction: Improving Balanced Random Forests

Łukasz Degórski
Institute of Computer Science
Polish Academy of Sciences
ul. J. K. Ordona 21
01-237 Warszawa, Poland
Email: ldegorski@bach.ipipan.waw.pl

Łukasz Kobyliński
Institute of Computer Science
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warszawa, Poland
Email: L.Kobylinski@elka.pw.edu.pl

Adam Przepiórkowski
Institute of Computer Science
Polish Academy of Sciences
ul. J. K. Ordona 21
01-237 Warszawa, Poland
Email: adamp@ipipan.waw.pl

***Abstract*—The article discusses methods of improving the ways of applying Balanced Random Forests (BRFs), a machine learning classification algorithm, used to extract definitions from written texts. These methods include different approaches to selecting attributes, optimising the classifier prediction threshold for the task of definition extraction and initial filtering by a very simple grammar.**

## I. Introduction

THE paper deals with extracting definitions from relatively unstructured instructive texts (textbooks, learning materials in eLearning, etc.) in a morphologically rich, relatively free word order, determinerless language (Polish). The same methods could easily be used for other similar languages without or with only minor changes, though. The work reported here is a continuation of work carried out within the recently finished *Language Technology for eLearning* project (LT4eL; http://www.lt4el.eu/).

The aim of the paper is to show how the results of previous attempts, presented in [1], can be improved by choosing the optimal threshold of classifier's prediction, with respect to the task of definition extraction, as well as to show that these improved results are close to optimal, in the sense that preliminary filtering by a simple grammar does not improve them significantly, as it was the case in experiments described in [2]. Attempts to use a different set of attributes will also be mentioned.

We used the same corpus of instructive texts as in [1] and [2]. It was automatically annotated morphosyntactically and then manually annotated for definitions, and contains over 30000 tokens, almost 11000 sentences and 558 definitions. These were divided by annotators into 6 types, depending on the most recognisable marker of being a definition:

- copula verb (e.g. cat **is a** domestic animal…)
- other verbs (e.g. we **define** a cat as a domestic animal…)
- punctuation (e.g. cat**:** a domestic animal…)
- layout (e.g. defined phrase in bold, large font, the definition in smaller font in the next line)
- pronoun
- other

We performed the experiments on two corpora: the whole set (described above) and its copula-type subset (the same sentences, but only 173 definitions). The experiments for other languages, conducted by other members of the LT4eL project, have shown that copula definitions have the highest probability of being successfully extracted by means of machine learning methods.

Note that the number of definitions in both sets is not exactly equal to the number of what we later call *definitional sentences*. Manually annotated definitions may begin or end in the middle of a sentence, and span multiple sentences. However, the ML methods operate on sentence level: a definitional sentence in this context is a sentence that has a nonempty intersection with at least one definition. For instance, in the whole set there are 546 definitional sentences.

The rest of the paper is organized as follows. In Section II we describe the classification method used for definition extraction. In Section III we discuss the possibilities of choosing representative attributes of words for the task of definition extraction. In Section IV we present differences in the achieved results, with respect to chosen methodology of interpreting classifier's outcome. In Section V we present the influence of manually constructed grammars on the accuracy of our definition extraction approach. Finally, we present the previous work done in the field in Section VI and conclude in Section VII.

## II. BRF algorithm

Random Forest (RF; [3]) is a homogeneous ensemble of unpruned decision trees (e.g., CART, C4.5; [4]), where—at each node of the tree—a subset of all attributes is randomly selected and the best attribute on which to further grow the tree is taken from that random set. Additionally, Random Forest is an example of the bagging (bootstrap aggregating) method, i.e., each tree is trained on a set bootstrapped from the original training set. Decisions are reached by simple voting.

Balanced Random Forest (BRF; [5]) is a modification of RF, where for each tree two bootstrapped sets of the same size, equal to the size of the minority class, are constructed: one for the minority class, the other for the majority class. Jointly, these two sets constitute the training set.

Similarly as in [1], for the task of extracting definitions from a set of documents by sentence classification, we use the following version of the BRF algorithm:

978-83-60810-14-9/08/$25.00 © 2008 IEEE

- split the training corpus into $n_d$ definitions and $n_{nd}$ non-definitions; the input data is heavily imbalanced, so $n_d \ll n_{nd}$;
- construct $k$ trees, each in the following way:
  - draw a bootstrap sample of size $n_d$ of definitions, and a bootstrap sample of the same size $n_d$ of non-definitions,
  - learn the tree (without pruning) using the CART algorithm, on the basis of the sum of the two bootstrap samples as the training corpus, but:
  - at each node, first select at random $m$ features (variables) from the set of all $M$ features ($m \ll M$; selection without replacement), and only then select the best feature (out of these $m$ features) for this node; this random selection of $m$ features is repeated for each node;
- the final classifier is the ensemble of the $k$ trees and decisions are reached by simple voting.

We have chosen the value of $m$ to be equal to $\sqrt{M}$ in all the experiments, although other sufficiently small values of $m$ could be used, as discussed in [3].

Up to $k = 800$ random trees were generated in each experiment. We always quote the results for the best-performing number of iterations in a given configuration (corpus, attributes, optimisation and filtering). The best-performing number varied between 300 and 700 for different configurations.

## III. Choosing the attributes

In [1], a set of 10 permutations of $n$-gram types was used for document representation as machine learning attributes (Table I). The set was carefully chosen by a half-statistical, half-heuristic method (having in mind the $\chi^2$ statistic value with respect to the class attribute and statistical independence of the attributes). In these experiments 100 most common $n$-grams of each of the 10 types were used for document representation, resulting in a dataset of ca. 900 binary attributes (fewer than 100 values for *ctag* unigrams exist) and 10830 instances. Data instances correspond to document sentences, while the values of binary attributes indicate whether a particular $n$-gram appears in the sentence.

TABLE I
The previously used set of $n$-gram types.

| no. | $n$-gram | | | no. | $n$-gram | | |
|---|---|---|---|---|---|---|---|
| 1 | *base* | | | 6 | *base* | *base* | |
| 2 | *ctag* | *ctag* | *case* | 7 | *ctag* | *ctag* | |
| 3 | *ctag* | *base* | | 8 | *ctag* | *case* | |
| 4 | *base* | *case* | | 9 | *base* | *base* | *base* |
| 5 | *base* | *ctag* | | 10 | *ctag* | | |

In our current experiments we have tried a slightly different method. For each of the possible 39 permutations of 1-grams, 2-grams and 3-grams of available features: *base* (base word form), *case* (grammatical case) and *ctag* (part of speech of the word), we generate up to 100 most frequent $n$-grams. As not for all permutations 100 different $n$-grams exist, the final set has around 3750 attributes.

In each iteration of 10-fold cross-validation we proceed as follows:

- in the training set (90% of the corpus):
  1) order the attributes according to the value of the $\chi^2$ statistic with respect to the class attribute,
  2) select the top 900 attributes (those fitting the example class best),
  3) train the Balanced Forest classifier on the set;
- in the test set (10% of the corpus):
  4) reject all attributes not on the top 900 list,
  5) apply the classifier.

The number of attributes was not chosen arbitrarily. Previous experiments (cf. Table 4 in [1]) have shown that increasing the number of $n$-grams of each of the selected types over 100 does not improve the classification results. That is the reason why in that method (with 10 $n$-gram types, and not all types had 100 $n$-grams) about 900 attributes were used. For comparability, in the new method we used a similar number of attributes – chosen differently though.

The experiments were performed on the whole set of definitions (as in [1]), and also on a version of the corpus in which only the copula definitions were marked. We have used the two known versions of the F measures to assess the results:

$$F_\alpha = \frac{(1+\alpha)\cdot(\text{precision}\cdot\text{recall})}{\alpha\cdot\text{precision}+\text{recall}}$$

$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{\beta^2\cdot\text{precision}+\text{recall}}$$

The new method gave promising results for the copula definitions:

TABLE II
Comparison of attribute selection methods, copula definitions

| attributes | precision | recall | $F_{\alpha=1}$ | $F_{\alpha=2}$ | $F_{\beta=2}$ | $F_{\alpha=5}$ |
|---|---|---|---|---|---|---|
| preselected | 16.50 | **84.40** | 27.60 | 35.59 | 46.30 | 50.06 |
| $\chi^2$ | **17.60** | 81.70 | **28.96** | **36.90** | **47.27** | **50.84** |

Unfortunately it turned out to be disappointing when applied to all definitions:

TABLE III
Comparison of attribute selection methods, all definitions

| attributes | precision | recall | $F_{\alpha=1}$ | $F_{\alpha=2}$ | $F_{\beta=2}$ | $F_{\alpha=5}$ |
|---|---|---|---|---|---|---|
| preselected | **21.37** | **69.04** | **32.64** | **39.60** | **47.74** | **50.33** |
| $\chi^2$ | 20.60 | 65.20 | 31.31 | 37.87 | 45.50 | 47.91 |

This leads to a conclusion that the more general method of choosing $n$-gram types for the task of definition extraction may still perform better than direct selection of specific $n$-grams in each classification iteration. The advantage of performing a purely statistical attribute selection lies in eliminating any preconceived notions about the role of certain word $n$-gram types in discriminating definitional sentences from non-definitional. On the other hand, a preselected set of $n$-gram types may be used without any further data analysis for document representation in other, similar problems, maybe even different languages.

## IV. Optimising the Threshold

The task of extracting definitions from an annotated corpus of documents was defined by the LT4eL project mentioned above, which focused on facilitating the construction and retrieval of learning objects (instructive material) in eLearning with the help of language technology. The results of automatic definition extraction were to be presented to the author or the maintainer of a learning object as candidates for the glossary of this learning object.

The intended use determines the appropriate approach. It is obviously easier to reject wrong definition candidates than to go back to the text and search for missed good definitions manually, so in this application recall was more important than precision. In [1] this assumption was exploited at the evaluation level only. $F_{\alpha=2}$ and $F_{\beta=2}$ were taken into account when comparing the approaches and datasets, to acknowledge the preference for recall. The classifier's prediction threshold of being a definition was set arbitrarily to 0.5 there. As Balanced Forest algorithm takes care of weighting the imbalanced number of examples of both classes (definitions and non-definitions), this approach does not favour any class, so the ratio of correctly classified examples to all examples was maximised.

However, it is worth noting that this is not exactly what we need here. Favouring recall over precision, we would like to focus more on the correctly classified positive examples, at the inevitable cost of misclassifying some of the negative ones. On the other hand, exactly how many times the recall is more important than precision in this case is an empirical issue. Answering this question would require user case evaluation experiments and as such is out of the scope of this article.

We have focused on maximising the $F_2$ measure, in two known approaches to its calculation, supposing recall is twice as important as precision.[1] Note that this *intended bias* towards recall has nothing to do with the imbalance of the classes in the training data (definitions vs. non-definitions). Thus, instead of maximising the ratio of correctly classified examples, we maximise the values of both $F_2$ measures by selecting the classification threshold appropriately. This means we may favour one of the classes over another, if this leads to an increase of the value of the chosen measure.

For the results, cf. Table IV and Table V. There is a clear improvement in terms of the chosen measures that can be explained by the accompanying four figures. The peaks of the graphs, especially those representing F-measures on the copula definition subcorpora, are located quite far to the right from 0.5 (that is, the default value used when there is no optimisation). However, we have to be well aware of the needs: fine-tuning the threshold value to one measure might also make the results with regard to other measures worse. On the other hand, both measures tend to peak close to each other (and not always close to 0.5). That may suggest that in case of an unknown corpus it is better to optimise with regard to a similar measure than not to optimise at all—as the graph for this corpus might peak far away from 0.5. The question what is a similar measure and what is not remains open though, and we will not attempt to address it in this paper.

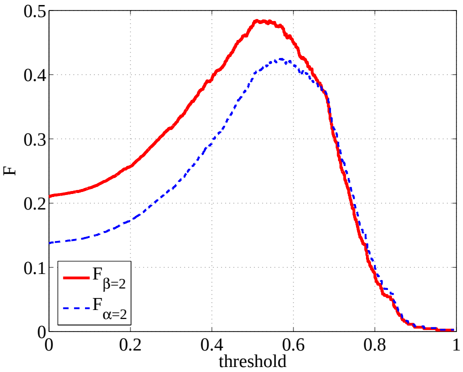


Fig. 1. F-measures values with respect to the chosen threshold on the dataset with all definitions and preselected set of attributes

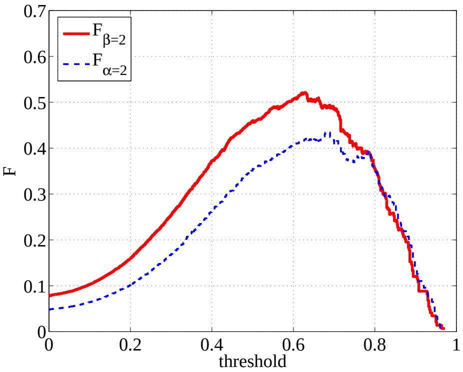


Fig. 2. F-measures values with respect to the chosen threshold on the dataset with copula definitions only and preselected set of attributes

## V. Applying a Manually Created Grammar

As described in [2], applying a very simple partial grammar before the classifiers such as Naïve Bayes, decision trees ID3 and C4.5, AdaBoostM1 with Decision Stump, Support Vector Machines and lazy classifier IB1 significantly improves the results. In that approach all sentences rejected by the grammar are unconditionally marked as non-definitions, and only those accepted by the grammar may be marked as definitions in the Machine Learning stage.

[1] There are different views in literature on how this should be done. For instance, [6] uses $F_\beta$, which is in fact the same formula as $F_\alpha$, but giving quadratic importance to the parameter instead of linear: $F_{\alpha=4} = F_{\beta=2}$. Something that could be interpreted as third version is used for instance in [7], but at a closer look it turns out to be effectively equivalent to $F_\alpha$ – used also in [2] and some medical papers – but encoding the intended result differently: $F_{0.5}$ is used to denote a measure giving equal weights to precision and recall (as $F_{\alpha=1}$), and $F_{0.75}$ is said to value recall three times more than precision (as $F_{\alpha=3}$).

TABLE IV
GAIN IN F-MEASURES FROM OPTIMISING THE THRESHOLD, ALL DEFINITIONS

| | preselected attributes | | | | $\chi^2$ attributes | | | |
|---|---|---|---|---|---|---|---|---|
| threshold | precision | recall | $F_{\alpha=2}$ | $F_{\beta=2}$ | precision | recall | $F_{\alpha=2}$ | $F_{\beta=2}$ |
| no optimisation | 21.37 | 69.04 | 39.60 | 47.74 | 20.60 | 65.20 | 37.87 | 45.50 |
| optimised for $F_{\alpha=2}$ | 27.80 | 57.69 | **42.47** | 47.48 | 26.38 | 55.13 | **40.44** | 45.26 |
| optimised for $F_{\beta=2}$ | 22.30 | 68.50 | 40.52 | **48.43** | 23.48 | 60.99 | 39.80 | **46.22** |

TABLE V
GAIN IN F-MEASURES FROM OPTIMISING THE THRESHOLD, COPULA DEFINITIONS ONLY

| | preselected attributes | | | | $\chi^2$ attributes | | | |
|---|---|---|---|---|---|---|---|---|
| threshold | precision | recall | $F_{\alpha=2}$ | $F_{\beta=2}$ | precision | recall | $F_{\alpha=2}$ | $F_{\beta=2}$ |
| no optimisation | 16.50 | 84.40 | 35.59 | 46.30 | 17.60 | 81.70 | 36.90 | 47.27 |
| optimised for $F_{\alpha=2}$ | 25.42 | 67.78 | **43.58** | 50.84 | 31.78 | 60.56 | **46.52** | 51.27 |
| optimised for $F_{\beta=2}$ | 22.68 | 77.22 | 42.86 | **52.14** | 28.26 | 65.00 | 45.35 | **51.59** |

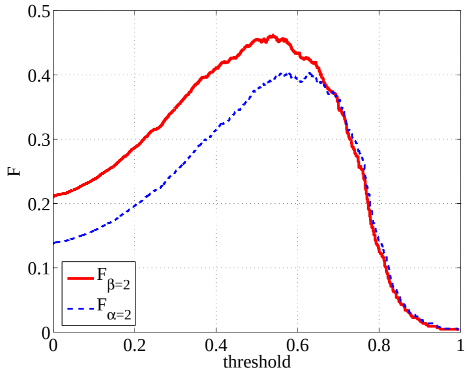


Fig. 3. F-measures values with respect to the chosen threshold on the dataset with all definitions and $\chi^2$ attribute selection

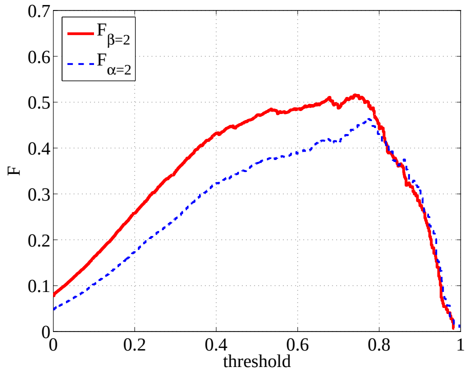


Fig. 4. F-measures values with respect to the chosen threshold on the dataset with copula definitions only and $\chi^2$ attribute selection

Even such a primitive grammar (that could also be described as a set of pattern-matching rules) rejected a significant part of potential false positives, i.e. those sentences that would be mistakenly marked as definitions by the classifiers. Thus, a significant relative increase of precision (for different classifiers from 36% up to 72%) was observed, accompanied only by a minor decrease of recall (between 3.4% and 7.5%). In terms of $F_{\alpha=2}$ measure, the increase was between 21% and 40%.

TABLE VI
THE RESULTS OF THE SIMPLE PARTIAL GRAMMAR BY ITSELF

| corpus | precision | recall | $F_{\alpha=2}$ | $F_{\beta=2}$ |
|---|---|---|---|---|
| whole | 9.10 | 89.60 | 22.69 | 32.36 |
| copula | 3.30 | 99.40 | 9.28 | 14.57 |

In case of the Balanced Random Forest classifier the gain turned out to be much smaller, up to 3.4%—cf. Table VII. Note that we look at the relative gain, not the absolute values of precision, recall and F-measures, because those numbers are not directly comparable: in [2] the experiments were not performed as a ten-fold cross-validation, but on a separate training and test subcorpora.

TABLE VII
GAIN OF APPLYING A SIMPLE GRAMMAR BEFORE THE CLASSIFIERS, ALL DEFINITIONS

| pre-filtering | $F_{\alpha=2}$ standard | $F_{\alpha=2}$ optimised | $F_{\beta=2}$ standard | $F_{\beta=2}$ optimised |
|---|---|---|---|---|
| no | 39.60 | 42.47 | 47.74 | 48.43 |
| yes | **40.95** | **43.09** | **48.62** | **49.30** |
| relative gain | 3.4% | 1.5% | 1.8% | 1.8% |

Balanced Random Forest classifier, especially with threshold optimisation, is inherently good enough not to require the initial pre-filtering by the grammar. We conclude that there is not that many potential false positives to be removed. This is clear when we look at the results for copula definitions in Table VIII.

TABLE VIII
GAIN OF APPLYING A SIMPLE GRAMMAR BEFORE THE CLASSIFIERS, COPULA DEFINITIONS

| pre-filtering | $F_{\alpha=2}$ standard | $F_{\alpha=2}$ optimised | $F_{\beta=2}$ standard | $F_{\beta=2}$ optimised |
|---|---|---|---|---|
| no | 35.59 | 43.58 | 46.30 | 52.14 |
| yes | **36.35** | **43.67** | **47.07** | **52.29** |
| relative gain | 2.1% | 0.2% | 1.7% | 0.3% |

## VI. PREVIOUS WORK

There is a substantial previous work on definition extraction, as this is a subtask of many applications, including terminology extraction [8], the automatic creation of glossaries [9], [10], question answering [11], [12], learning lexical semantic relations [13], [14] and the automatic construction of ontologies [15]. Despite the current dominance of the ML paradigm in NLP, tools for definition extraction are invariably language-specific and involve shallow or deep processing, with most work done for English and other Germanic languages, as well as French.

For Polish, first attempts at constructing definition extraction systems are described—in the context of other Slavic languages—in [16], and improved results are presented in [17]. [2] describes improvements achieved by using a simple manually created grammar.

The first NLP applications of the plain Random Forests are apparently those reported in [18] and in [19], where they are used in the classical language modelling task (predicting a sequence of words) for speech recognition and give better results than the usual $n$-gram based approaches.

The use of Balanced Random Forests for definition extraction in textual datasets was proposed in [1].

## VII. Conclusion

For definition extraction, the Balanced Random Forest classification method may be further improved by optimising the threshold above which we classify a given sentence as a definition. With this improvement, the algorithm does not gain much more from initial filtering of the data by a very simple, high-recall hand-crafted grammar, as it was in case of other ML classifiers we experimented with; however, the gain, being small, is always positive, so it may be worth trying, when the best possible result is desired, even at the cost of complicating the algorithm and lengthening the execution time. The same applies to using a more advanced set of attributes that are selected for each training set separately instead of using a preselected single set.

## References

[1] Ł. Kobyliński and A. Przepiórkowski, "Definition extraction with balanced random forests," in *Proceedings of the 6th International Conference on Natural Language Processing, GoTAL 2008, Gothenburg, Sweden*, ser. Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag, 2008.

[2] Ł. Degórski, M. Marcińczuk, and A. Przepiórkowski, "Definition extraction using a sequential combination of baseline grammars and machine learning classifiers," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008*. ELRA, 2008.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[4] J. R. Quinlan, *Programs for Machine Learning*. Morgan Kaufmann, 1993.

[5] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," University of California, Berkeley, Tech. Rep. 666, 2004, http://www.stat.berkeley.edu/tech-reports/666.pdf.

[6] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2008.

[7] M. Jansche, "Maximum expected f-measure training of logistic regression models," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*. Vancouver: ACL, 2005, pp. 692–699.

[8] J. Pearson, "The expression of definitions in specialised texts: a corpus-based analysis," in *Proceedings of the Seventh Euralex International Congress*, M. Gellerstam, J. Järborg, S. G. Malmgren, K. Norén, L. Rogström, and C. Papmehl, Eds., Göteborg, 1996, pp. 817–824.

[9] J. L. Klavans and S. Muresan, "DEFINDER: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text," in *Proceedings of the Annual Fall Symposium of the American Medical Informatics Association*, 2000.

[10] ——, "Evaluation of the DEFINDER system for fully automatic glossary construction," in *Proceedings of AMIA Symposium 2001*, 2001.

[11] S. Miliaraki and I. Androutsopoulos, "Learning to identify single-snippet answers to definition questions," in *Proceedings of COLING 2004*, Geneva, Switzerland, 2004, pp. 1360–1366.

[12] I. Fahmi and G. Bouma, "Learning to identify definitions using syntactic features," in *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*, 2006.

[13] V. Malaisé, P. Zweigenbaum, and B. Bachimont, "Detecting semantic relations between terms in definitions," in *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, S. Ananadiou and P. Zweigenbaum, Eds., Geneva, Switzerland, 2004, pp. 55–62.

[14] A. Storrer and S. Wellinghoff, "Automated detection and annotation of term definitions in German text corpora," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006*. Genoa: ELRA, 2006.

[15] S. Walter and M. Pinkal, "Automatic extraction of definitions from German court decisions," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006, pp. 20–28. [Online]. Available: http://www.aclweb.org/anthology/W/W06/W06-0203

[16] A. Przepiórkowski, Ł. Degórski, M. Spousta, K. Simov, P. Osenova, L. Lemnitzer, V. Kuboň, and B. Wójtowicz, "Towards the automatic extraction of definitions in Slavic," in *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing at ACL 2007*, J. Piskorski, B. Pouliquen, R. Steinberger, and H. Tanev, Eds., 2007, pp. 43–50.

[17] A. Przepiórkowski, Ł. Degórski, and B. Wójtowicz, "On the evaluation of Polish definition extraction grammars," in *Proceedings of the 3rd Language & Technology Conference*, Z. Vetulani, Ed., Poznań, Poland, 2007, pp. 473–477.

[18] R. D. Nielsen and S. Pradhan, "Mixing weak learners in semantic parsing," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, D. Lin and D. Wu, Eds. Barcelona: ACL, 2004, pp. 80–87.

[19] P. Xu and F. Jelinek, "Random forests in language modeling," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, D. Lin and D. Wu, Eds. Barcelona: ACL, 2004, pp. 325–332.

[20] D. Lin and D. Wu, Eds., *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*. Barcelona: ACL, 2004.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# An application supporting language analysis within the framework of the phonetic grammar

Krzysztof Dyczkowski
Adam Mickiewicz University
Faculty of Mathematics and Computer Science,
Umultowska 87, 61-614 Poznań, Poland
Email: chris@amu.edu.pl

Norbert Kordek, Paweł Nowakowski, Krzysztof Stroński
Adam Mickiewicz University,
Institute of Linguistics,
al. Niepodległości 4, 61-874 Poznań, Poland
Email: {norbert, gpn, stroniu}@amu.edu.pl

***Abstract*—The aim of the paper is to present an application supporting language analysis within the framework of the phonetic grammar. The notion of the phonetic grammar has been concisely introduced and the basic potential of the application and the algorithms employed in it are briefly discussed. The application is to enable a uniform description and a comparative analysis of many languages. At the first stage the languages taken into consideration are Polish, Chinese and Hindi.**

## I. The phonetic grammar

### A. Introduction

PHONETICS is a field of linguistics which is concerned with the articulatory, acoustic, auditory and distributive properties of phones. Phonetics of a given language is also sometimes understood as a set of phones relevant to a given language (e.g. the phonetics of Polish language). The phone is a set of all hic et nunc pronounced homophonous speech sounds. The speech sounds are of temporal character and their number is actually infinite. To reduce the number of elements belonging to hic and nunc pronounced speech sounds we classify them into sets of phones e.g. the set of all homophonous temporal realizations of the speech sounds p1, p2, p3, p4, .... is considered to be the phone [p]. All phones are described in terms of the articulatory features. E.g. the relevant features of [p] are: voiceless, oral, hard, plosive, labial etc. Assigning an exhaustive feature set to a given phone is equal with the definition of the phone.

Phonetic grammar is understood as a set of relations between articulatory features and articulatory dimensions (sets of homogenous articulatory features). The concept is based on the theory introduced in the works of Jerzy Bańczerowski ([1], [2]).

The aims of the present project are as follows:

- elaboratation of a uniform (for each language) set of articulatory features,
- elaboratation of a mathematical model of the relations between languages,
- preparation of computer tools for the processing of the collected data, i.e. to elaborate a model of the linguistic data collecting,
- implementation of algorithms of the data processing,
- preparation of algorithms of comparative analysis of languages using the mathematical model,
- comparative analysis of the phonetic grammars of Polish, Chinese and Hindi.

To enable the comparative analyses of languages we introduce a uniform description of the phones of a given language as a set of articulatory features belonging to one of the following articulatory dimensions:

- the mechanism of the air flow origin,
- the direction of the air flow,
- the state of glottis,
- the way of air flow
- the place of articulation,
- the articulator,
- the degree of supraglottal aperture,
- the vertical position of the tongue—the horizontal position of the tongue,
- the degree of labialization,
- the degree of delabialization,
- the duration of articulation
- the degree of supra- and subglottal tension,
- the frequency of articulatory approximation.

### B. Phones as objects in $n$-dimensional space

The original method introduced the notion of the articulatory distance between phones (see [2]). The distance is interpreted as a number of differential features (features by which given phones differ from others). It is thus reducible to the well known Hamming distance.

Our team has proposed to introduce numerical interpretation of the articulatory dimensions. Let $G$ be a set of all phones within which the subsets $G_l$ of the phones belonging to a given language can be specified (where $l$ is an index of a given language) and let $W = \{W_1, W_2, \ldots, W_n\}$ be a set of articulatory dimensions, where $n$ is a number of articulatory dimensions. Thus each phone $g$ from the set $G$ is specified by the vector in $n$-dimensional metric space $\mathbb{R}^n$. Each articulatory feature is uniquely specified by one numerical value from the interval $[0, k]$, where $k$ is a maximal number of features in a dimension. Ascribing a proper numerical value to the feature mirrors the natural order of the features in a given dimension.

The research project is supported by Ministry of Science and Higher Education grant N N104 327434.

Thus each phone $g = (c_1, c_2, \ldots, c_n)$ where $c_i$ belongs to the set of features of a given dimension $W_i$.

The notion of the phone as a point in $n$-dimensional space enables application of well known measures of distances. For example for the pair of phones $a, b \in G$ we can specify the following measures of distances:

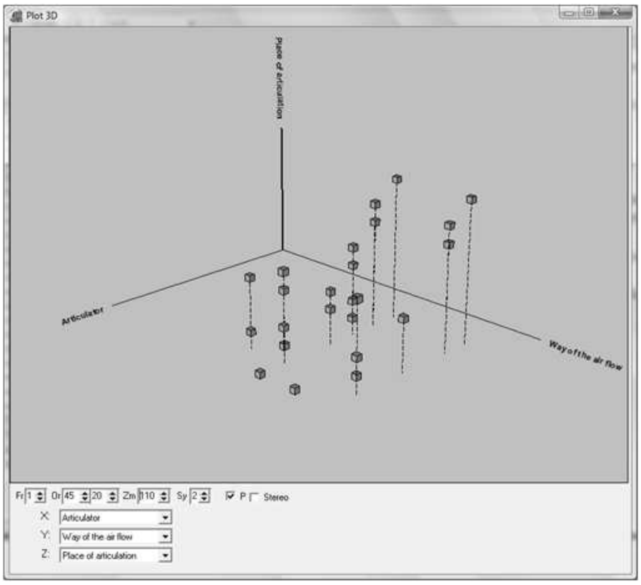
- The Minkowski distance for $m \geq 1$:
$$Dist_M(a,b) = \left( \sum_{i=1}^{n} |a_i - b_i|^m \right)^{1/m}$$
- The Manhattan distance:
$$Dist_H(a,b) = \sum_{i=1}^{n} |a_i - b_i|$$
being a particular instance of the Minkowski distance for $m = 1$,
- The Euclidean distance:
$$Dist_E(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$
being a particular instance of the Minkowski distance $m = 2$.

The distances defined in this manner will enable us to build similarity measures between phones and between phonetic systems of given languages. We assume that sound more distant from each other in the sense of the appropriate metrics are less similar to each other. And this seems to be in accordance with the intuition.

## II. Computer application

### A. The tool for collecting phone inventories

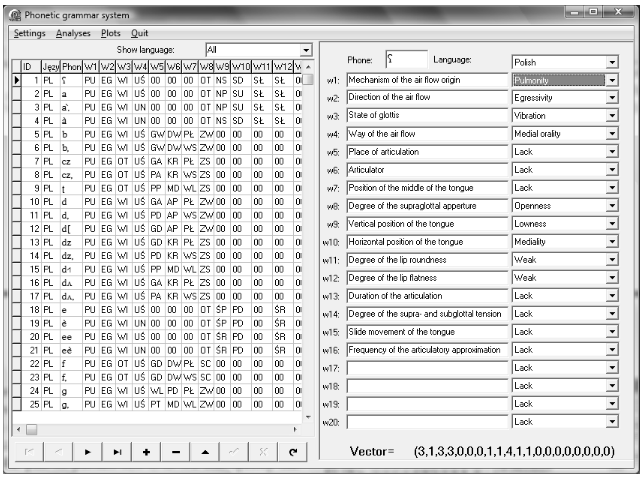
The first essential element in the system has been to build a database and a relevant interface enabling data insertion using the standardized International Phonetic Alphabet (IPA).



Fr 1 Or 45 20 Zm 110 Sy 2 ☑ P ☐ Stereo

X: Articulator
Y: Way of the air flow
Z: Place of articulation

Fig. 1. Phones in selected 3 dimensions



Show language: All

| | ID | Języ | Phon | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | W11 | W12 | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ▶ | 1 | PL | ʕ | PU | EG | WI | UŚ | 00 | 00 | 00 | OT | NS | SD | SŁ | SŁ | 0 |
| | 2 | PL | a | PU | EG | WI | UŚ | 00 | 00 | 00 | OT | NP | SU | SŁ | SŁ | 0 |
| | 3 | PL | à, | PU | EG | WI | UN | 00 | 00 | 00 | OT | NP | SU | SŁ | SŁ | 0 |
| | 4 | PL | à | PU | EG | WI | UN | 00 | 00 | 00 | OT | NS | SD | SŁ | SŁ | 0 |
| | 5 | PL | b | PU | EG | WI | UŚ | GW | DW | PŁ | ZW | 00 | 00 | 00 | 00 | 0 |
| | 6 | PL | b, | PU | EG | WI | UŚ | GW | DW | WS | ZW | 00 | 00 | 00 | 00 | 0 |
| | 7 | PL | cz | PU | EG | OT | UŚ | GA | KR | PŁ | ZS | 00 | 00 | 00 | 00 | 0 |
| | 8 | PL | cz, | PU | EG | OT | UŚ | PA | KR | WS | ZS | 00 | 00 | 00 | 00 | 0 |
| | 9 | PL | ţ | PU | EG | OT | UŚ | PP | MD | WL | ZS | 00 | 00 | 00 | 00 | 0 |
| | 10 | PL | d | PU | EG | WI | UŚ | GA | AP | PŁ | ZW | 00 | 00 | 00 | 00 | 0 |
| | 11 | PL | d, | PU | EG | WI | UŚ | PD | AP | WS | ZW | 00 | 00 | 00 | 00 | 0 |
| | 12 | PL | d[ | PU | EG | WI | UŚ | GD | AP | PŁ | ZW | 00 | 00 | 00 | 00 | 0 |
| | 13 | PL | dz | PU | EG | WI | UŚ | GD | KR | PŁ | ZS | 00 | 00 | 00 | 00 | 0 |
| | 14 | PL | dz, | PU | EG | WI | UŚ | PD | KR | WS | ZS | 00 | 00 | 00 | 00 | 0 |
| | 15 | PL | d˦ | PU | EG | WI | UŚ | PP | MD | WL | ZS | 00 | 00 | 00 | 00 | 0 |
| | 16 | PL | dʌ | PU | EG | WI | UŚ | GA | KR | PŁ | ZS | 00 | 00 | 00 | 00 | 0 |
| | 17 | PL | dʌ, | PU | EG | WI | UŚ | PA | KR | WS | ZS | 00 | 00 | 00 | 00 | 0 |
| | 18 | PL | e | PU | EG | WI | UŚ | 00 | 00 | 00 | OT | ŚP | PD | 00 | ŚR | 0 |
| | 19 | PL | è | PU | EG | WI | UN | 00 | 00 | 00 | OT | ŚP | PD | 00 | ŚR | 0 |
| | 20 | PL | ee | PU | EG | WI | UŚ | 00 | 00 | 00 | OT | ŚR | PD | 00 | ŚR | 0 |
| | 21 | PL | eè | PU | EG | WI | UN | 00 | 00 | 00 | OT | ŚR | PD | 00 | ŚR | 0 |
| | 22 | PL | f | PU | EG | OT | UŚ | GD | DW | PŁ | SC | 00 | 00 | 00 | 00 | 0 |
| | 23 | PL | f, | PU | EG | OT | UŚ | GD | DW | WS | SC | 00 | 00 | 00 | 00 | 0 |
| | 24 | PL | g | PU | EG | WI | UŚ | WL | PD | PŁ | ZW | 00 | 00 | 00 | 00 | 0 |
| | 25 | PL | g, | PU | EG | WI | UŚ | PT | MD | WL | ZW | 00 | 00 | 00 | 00 | 0 |

Phone: ʕ Language: Polish

| | | |
|---|---|---|
| w1: | Mechanism of the air flow origin | Pulmonity |
| w2: | Direction of the air flow | Egressivity |
| w3: | State of glottis | Vibration |
| w4: | Way of the air flow | Medial orality |
| w5: | Place of articulation | Lack |
| w6: | Articulator | Lack |
| w7: | Position of the middle of the tongue | Lack |
| w8: | Degree of the supraglottal apperture | Openness |
| w9: | Vertical position of the tongue | Lowness |
| w10: | Horizontal position of the tongue | Mediality |
| w11: | Degree of the lip roundness | Weak |
| w12: | Degree of the lip flatness | Weak |
| w13: | Duration of the articulation | Lack |
| w14: | Degree of the supra- and subglottal tension | Lack |
| w15: | Slide movement of the tongue | Lack |
| w16: | Frequency of the articulatory approximation | Lack |
| w17: | | Lack |
| w18: | | Lack |
| w19: | | Lack |
| w20: | | Lack |

Vector= (3,1,3,3,0,0,0,1,1,4,1,1,0,0,0,0,0,0,0,0)

Fig. 2. Inventory of phones

The application enables:

- defining dimensions and features occurring in them,
- ascribing proper numerical values to the dimensions,
- defining the number of languages,
- introducing a repertoire of phones of a given language and the description of the phones in terms of the relevant set of articulatory features.

### B. Basic analyses

The application is to generate the data concerning detailed levels of analysis in the phonetic grammar of each of the analyzed languages:

- the phone articulemization[1],
- the combining of the articulatory features,
- the articulatory opposition and similarity of phones,
- differential and identifying articulatory dimensions,
- the articulatory distance and proximity.

The computer application will automatically generate:

- the articulatory distance of any two phones in a given language,
- the articulatory similarity of any two phones in a given language,
- the articulatory features of a given phone,
- the articulatory category of a given articulatory feature,
- the dimensions in which given phones differ,
- the dimensions in which given phones are identical,
- the set of phones which have a specified articulatory distance,
- the set of phones which have specified articulatory features,
- the combination of a given set of articulatory features,
- the average articulatory distance between phones,
- the most numerous articulatory category specified by a given number of features,

[1] The operation of articulemization consists in ascribing the articulatory characteristics to a given phone.

- the least numerous articulatory category specified by a given number of features,
- the set of relevant features discerning at least one pair of sounds,
- the number of pairs of phones being discerned by particular features,
- the number of pairs of phones being discerned by particular sets of features,
- the most frequently combined articulatory features in a given articulatory distance.

### C. *Applied data-mining algorithms*

The analyses presented in the last section are the basis of language analysis. They apply rudimentary statistical and combinatorics methods. In the present section we are going to explore methods from the data-mining domain which will allow to discover automatically new interdependencies between phones. It will in turn enable to show certain relations between languages which have been so far unnoticed. All algorithms applied here use measures of distances as measure of similarity between phones. These algorithms can be used for phones from one ore more languages.

*1)* **K-means algorithm** *([9],[12]):* The first of the algorithms requires an input of expected number of phone clusters. It allows to divide the phone inventory into particular number of disjunctive classes. For example put $k := 2$ results in the division of the set of phones into vowels and consonants.

The algorithm is composed of the following steps:

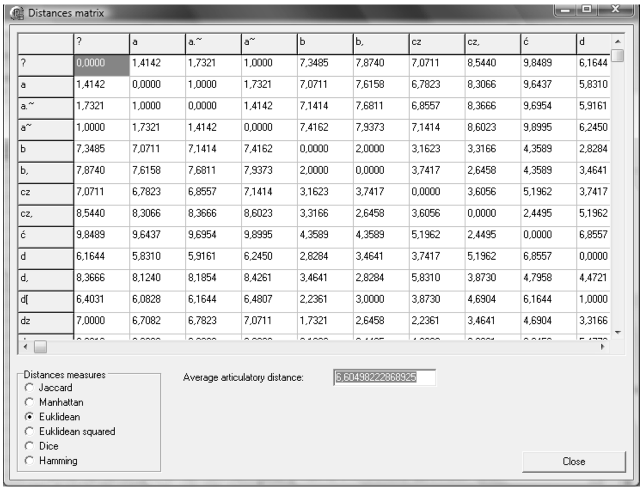
1. Place K points into the space represented by the phones that are being clustered. These points represent initial group centroids.
2. Assign each phone to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the $K$ centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

*2)* **The connected subgraphs algorithm** *([5],[7]):* This algorithm does not require an input of the number of clusters. It finds them itself on the basis of regularities in the data.

The algorithm operates on the basis of the matrix of distance $D$. It is a symmetric matrix $n \times n$ in which on the intersection of the columns and verses one receives the distance between the proper pair of phones and on the diagonal zero.

The algorithm operates in the following steps:

1. The distance matrix $D$ is calculated using the fixed distance.
2. Below the fixed threshold $\alpha$ all values in the matrix $D$ are zeroed. Finding the threshold is the basic element of the algorithm. In the simplest case it can be fixed as an average distance in the phone inventory reduced by the standard deviation of the average distance. The choice of the proper threshold mirrors our understanding how big the distance between phones must be to consider them too distant to be the members of the same group.



| | ? | a | a.~ | a~ | b | b, | cz | cz, | ć | d |
|---|---|---|---|---|---|---|---|---|---|---|
| ? | 0,0000 | 1,4142 | 1,7321 | 1,0000 | 7,3485 | 7,8740 | 7,0711 | 8,5440 | 9,8489 | 6,1644 |
| a | 1,4142 | 0,0000 | 1,0000 | 1,7321 | 7,0711 | 7,6158 | 6,7823 | 8,3066 | 9,6437 | 5,8310 |
| a.~ | 1,7321 | 1,0000 | 0,0000 | 1,4142 | 7,1414 | 7,6811 | 6,8557 | 8,3666 | 9,6954 | 5,9161 |
| a~ | 1,0000 | 1,7321 | 1,4142 | 0,0000 | 7,4162 | 7,9373 | 7,1414 | 8,6023 | 9,8995 | 6,2450 |
| b | 7,3485 | 7,0711 | 7,1414 | 7,4162 | 0,0000 | 2,0000 | 3,1623 | 3,3166 | 4,3589 | 2,8284 |
| b, | 7,8740 | 7,6158 | 7,6811 | 7,9373 | 2,0000 | 0,0000 | 3,7417 | 2,6458 | 4,3589 | 3,4641 |
| cz | 7,0711 | 6,7823 | 6,8557 | 7,1414 | 3,1623 | 3,7417 | 0,0000 | 3,6056 | 5,1962 | 3,7417 |
| cz, | 8,5440 | 8,3066 | 8,3666 | 8,6023 | 3,3166 | 2,6458 | 3,6056 | 0,0000 | 2,4495 | 5,1962 |
| ć | 9,8489 | 9,6437 | 9,6954 | 9,8995 | 4,3589 | 4,3589 | 5,1962 | 2,4495 | 0,0000 | 6,8557 |
| d | 6,1644 | 5,8310 | 5,9161 | 6,2450 | 2,8284 | 3,4641 | 3,7417 | 5,1962 | 6,8557 | 0,0000 |
| d, | 8,3666 | 8,1240 | 8,1854 | 8,4261 | 3,4641 | 2,8284 | 5,8310 | 3,8730 | 4,7958 | 4,4721 |
| d[ | 6,4031 | 6,0828 | 6,1644 | 6,4807 | 2,2361 | 3,0000 | 3,8730 | 4,6904 | 6,1644 | 1,0000 |
| dz | 7,0000 | 6,7082 | 6,7823 | 7,0711 | 1,7321 | 2,6458 | 2,2361 | 3,4641 | 4,6904 | 3,3166 |

Distances measures
○ Jaccard
○ Manhattan
● Euklidean
○ Euklidean squared
○ Dice
○ Hamming

Average articulatory distance: 6,60498222868925

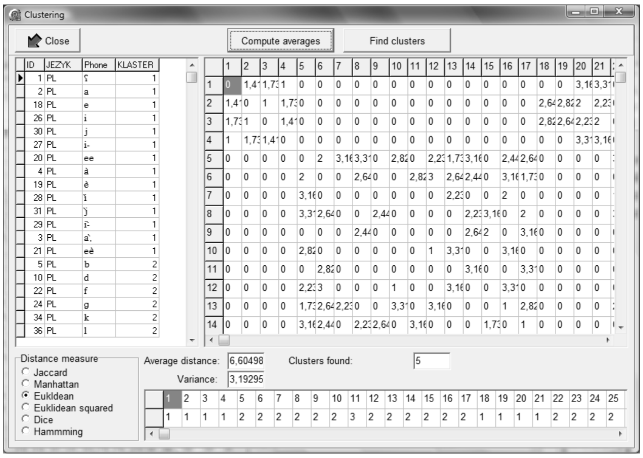
Fig. 3. The distances matrix for the Euclidean metric

3. Such a matrix is treated as a directed weighted graph in which non-zero values will mark weighted edges between the phones—the vertices. (also called threshold graph)
4. The algorithm Depth-first search (DFS) is applied. The algorithm results in finding connected subgraphs.
The subgraphs are wanted clusters.

*3)* **Agglomerative hierarchical clustering and dendrograms** *(see [10],[11]):* An agglomerative hierarchical clustering procedure produces a series of partitions of the data, $P_k, P_{k-1}, \ldots, P_1$. The first $P_k$ consists of $k$ single phone clusters, the last $P_1$, consists of a single group containing all $k$ phones.

At each particular stage the method joins together the two clusters which are closest together (most similar). At the first stage, this amounts to joining together the two objects that are closest together, since at the initial stage each cluster has one phone.

There are some different methods that use different ways of defining distance (or similarity) between clusters.



| ID | JEZYK | Phone | KLASTER |
|---|---|---|---|
| 1 | PL | ʕ | 1 |
| 2 | PL | a | 1 |
| 18 | PL | e | 1 |
| 26 | PL | i | 1 |
| 30 | PL | j | 1 |
| 27 | PL | i- | 1 |
| 20 | PL | ee | 1 |
| 4 | PL | à | 1 |
| 19 | PL | è | 1 |
| 28 | PL | ĭ | 1 |
| 31 | PL | j̆ | 1 |
| 29 | PL | ĭ- | 1 |
| 3 | PL | a`, | 1 |
| 21 | PL | eè | 1 |
| 5 | PL | b | 2 |
| 10 | PL | d | 2 |
| 22 | PL | f | 2 |
| 24 | PL | g | 2 |
| 34 | PL | k | 2 |
| 36 | PL | l | 2 |

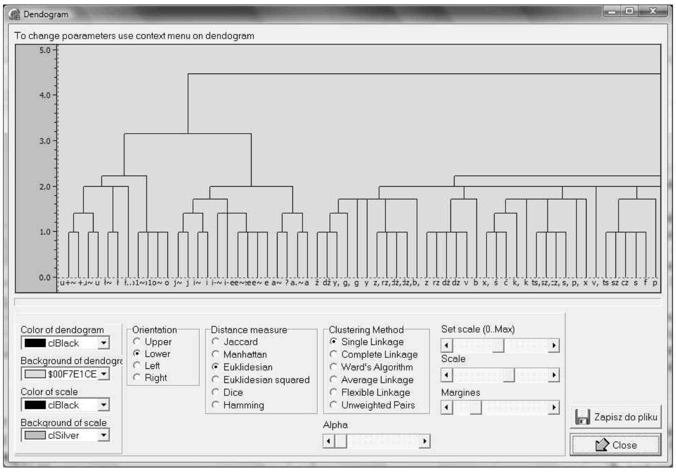
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1,41 | 1,73 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,16 | 3,31 |
| 2 | 1,41 | 0 | 1 | 1,73 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,64 | 2,82 | 2 | 2,23 |
| 3 | 1,73 | 1 | 0 | 1,41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,82 | 2,64 | 2,23 | 2 |
| 4 | 1 | 1,73 | 1,41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,31 | 3,16 |
| 5 | 0 | 0 | 0 | 0 | 0 | 2 | 3,16 | 3,31 | 0 | 2,82 | 0 | 2,23 | 1,73 | 3,16 | 0 | 2,44 | 2,64 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 2,64 | 0 | 0 | 2,82 | 3 | 2,64 | 2,44 | 0 | 3,16 | 1,73 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 3,16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,23 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 3,31 | 2,64 | 0 | 0 | 2,44 | 0 | 0 | 0 | 0 | 2,23 | 3,16 | 0 | 2 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2,44 | 0 | 0 | 0 | 0 | 0 | 2,64 | 2 | 0 | 3,16 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 2,82 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3,31 | 0 | 0 | 3,16 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 2,82 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3,16 | 0 | 0 | 3,31 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 2,23 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 3,16 | 0 | 0 | 3,31 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 1,73 | 2,64 | 2,23 | 0 | 0 | 3,31 | 0 | 3,16 | 0 | 0 | 0 | 1 | 2,82 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 3,16 | 2,44 | 0 | 2,23 | 2,64 | 0 | 3,16 | 0 | 0 | 0 | 1,73 | 0 | 1 | 0 | 0 | 0 | 0 |

Distance measure
○ Jaccard
○ Manhattan
● Eukldean
○ Euklidean squared
○ Dice
○ Hammming

Average distance: 6,60498
Variance: 3,19295
Clusters found: 5

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |

Fig. 4. The effect of the operation of the connected subgraphs algorithm



Dendogram

To change poarameters use context menu on dendogram

5.0
4.0
3.0
2.0
1.0
0.0
u+~ +ɹ~ u ł~ ł ł..)1~)1o~ o j~ j i~ i i-~ i-ee~:ee~ e a~ ? a.~a ź dź y, g, g y z, rz, dż, dz, b, z rz dż dz v b x, ś ć k, k ts, sz, cz, s, p, x v, ts sz cz s f p

Color of dendogram: clBlack

Background of dendogra: $00F7E1CE

Color of scale: clBlack

Background of scale: clSilver

Orientation: ○ Upper ● Lower ○ Left ○ Right

Distance measure: ○ Jaccard ○ Manhattan ● Euklidesian ○ Euklidesian squared ○ Dice ○ Hamming

Clustering Method: ● Single Linkage ○ Complete Linkage ○ Ward's Algorithm ○ Average Linkage ○ Flexible Linkage ○ Unweighted Pairs

Fig. 5. The example of the dendrogram

1. **Single linkage clustering**: It is one of the simplest agglomerative hierarchical clustering methods. It is also known as the nearest neighbor technique. The defining feature of the method is that distance between groups is defined as the distance between the closest pair of objects, where only pairs consisting of one object from each group are considered.
2. **Complete linkage clustering**: It is also called farthest neighbor, clustering method is the opposite of single linkage. Distance between groups is now defined as the distance between the most distant pair of objects, one from each group.
3. **Average linkage clustering**: Here the distance between two clusters is defined as the average of distances between all pairs of objects, where each pair is made up of one object from each group.
4. **Average group linkage**: With this method, groups once formed are represented by their mean values for each variable, that is, their mean vector, and inter-group distance is now defined in terms of distance between two such mean vectors.

The result of the operation of the algorithm is presented in the dendrogram. It is a special type of the dendric structure which assures an easy way of presenting the results of the hierarchical grouping.

Cutting the dendrogram at the selected level we can receive a proper division into a particular number of the groups of phones.

## III. Summary

The paper has presented the first stage of the realization of the more complex project which is meant to apply computer methods for the purpose of linguistic analyses.

At the next stages, besides the interpretation of the results we are going to apply the methods of fuzzy sets (mainly the notion of the linguistic variable) for the description of the repertoires of phones. The methodology of the linguistic summarization as a tool of the data analysis also seems to be very promising.

The results can be further used in different linguistic disciplines (also applied linguistics), especially in teaching foreign languages, speech analysis and in basic research on natural languages (in theory of linguistics and literary phonostylistics, comparative linguistics, typology).

## References

[1] J. Bańczerowski, “Phonetic Relations in the Perspective of Phonetic Dimensions“, *In: Pieper U., Stickel G. (eds.) Studia Linguistica Diachronica et Synchronica*. Berlin, 1985

[2] J. Bańczerowski, J. Pogonowski, T. Zgółka, “Wstęp do językoznawstwa.“ UAM, Poznań.

[3] T. Benni, “Fonetyka opisowa języka polskiego“, Ossolineum, Wrocław, 1964.

[4] C.K. Bhatia, “Consonant sequences in Standard Hindi“, Indian Linguistics, 1964, 25.206-12.

[5] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. *Lecture Notes in Computer Science, Di Battista and U. Zwick (Eds.)* :568–579, 2003.

[6] Yuen-ren. Chao, “A Grammar of Spoken Chinese“, University of California Press, Berkeley Ľ Los Angeles Ľ London, 1968.

[7] S. van Dongen, “Graph Clustering by Flow Simulation“, PhD thesis, University of Utrecht, 2000.

[8] L. Dukiewicz, T. Sawicka, “Fonetyka i fonologia“, W: *Urbańczyk S. (red.) Gramatyka współczesnego języka polskiego*, IJP PAN, Kraków, 1995.

[9] J. Han, M. Kamber, “Data Mining: Concepts and Techniques“, Morgan Kaufman, 2000.

[10] J. Hand, H. Mannila, P. Smyth, “Pricinciples of Data Mining“, MIT Press, 2001.

[11] A. Jain and R. Dubes,“Algorithms for Clustering Data“,Prentice-Hall, 1988.

[12] D.T. Larose,“Discovering Knowledge in Data: An Introduction to Data Mining“, Wiley, 2005.

[13] M. Ohala, “Aspects of Hindi Phonology“, Motilal Banarisidas, Delhi, 1983.

[14] M. Steffen-Batóg, “Studies in Phonetic Algorithms“, Sorus, Poznań, 1997.

[15] M. Steffen-Batóg, T. Batóg, “A Distance Function in Phonetics“, Lingua Posnaniensis 23, 47 Ľ 58, 1980.

[16] B. Wierzchowska B. “Opis fonetyczny języka polskiego“, PWN, Warszawa, 1967.

[17] Qin. Zhong, “On Chinese Phonetics“, Commercial Press, Beijing, 1980.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Set of Active Suffix Chains and its Role in Development of the MT System for Azerbaijani

Rauf Fatullayev
National E-Governance Project
Z.Aliyeva str. 33, AZ 1000, Baku,
Azerbaijan
Email: fatullayev@gmail.com

Ali Abbasov
National Academy of Science H.-
Cavid str., 31, AZ1143, Baku,
Azerbaijan
Email: ali@dcacs.ab.az

Abulfat Fatullayev
Institute of Linguistics of National
Academy of Science
H.Cavid str., 31, AZ1143, Baku,
Azerbaijan
Email: fabo@box.az

***Abstract*— Definition process of the active suffix chains of Azerbaijani (The Azerbaijani language) has been explained in this paper. Morphologically Azerbaijani word-forms consist of the stem and simple or compound suffixes (suffix chains). Because Azerbaijani is an agglutinative language the simple suffixes in this language can form the great number of suffix chains and consequently, it is possible to generate practically "endless" number of word-forms from the same stem. While developing machine translation system from Azerbaijani into non-Turkic languages for the translation of any word-form it is necessary to translate its suffix chain in whole. It is not possible to get the correct translation of a word-form by translating each simple suffix of its suffix chain separately and then by putting these translations together. For these reasons, it is necessary to define the subset of suffix chains frequently used in the texts instead of the set of every possible suffix chains.**

## I. Introduction

LANGUAGES of Turkic group (modern Turkish, Azerbaijani, Kazakh, Uzbek, Turkmen, Tatar, Kirghiz and others) are the morphologically rich languages and these languages are characterized by highly productive morphological processes that may produce a very large number of word forms for a given stem. Modeling each word-form as a separate lexical unit leads to a number of problems for the development of formal linguistic technologies as machine translation, speech recognition, text to speech etc. systems.

R esearch into the development of the machine translation (MT) systems for the languages of Turkic group is being carried out in two directions:

1. Development of the MT systems among languages of Turkic group;
2. Development of the MT systems from Turkic languages into languages not belonging to this group and vice versa;

While developing MT systems fro m the languages of Turkic group some problems related to the agglutinative nature of these languages arise. In Turkic languages, every word-form morphologically consists of the stem and the simple or compound suffixes (hereinafter we will call compound suffixes the suffix chains) and as mentioned above, it is possible to generate practically "endless" number of word-forms from the same stem.

Vocabul ary differences are prevalent for the languages of Turkic group. Since the morphological (for example: similar rules of the word formation, existence of the simple suffixes with the similar function) and syntactic structures (for example: the similar formation ways of the noun and verb phrases, the same word order) are very close, it is easier to develop machine translation systems of the first type than those of the second type.

Because the grammatical structures of these languages are very close, it is possible to develop an MT system by creating the Turkic-multilingual dictionary (Table A) of stems and the database of the equivalency (Table B) of simple suffixes [1]-[3].

For example, we presented the translation of the sentence "My little son goes to school" in six Turkic languages.

*Mən-im kiçik oğl-um məktəb-ə ged-ir* (Azerb.),
*Ben-im küçük oğl-um okul-a gid-iyor* (Turkish),
*Men-inq kiçik ўгл - uм maktab-qa bora-di* (Uzbek),
*Men-in bəlekey bala-m mektep-ke bara jat-ır* (Kazakh),
*Men-in kiçine uul-um mektep-ke bar-dı* (Kyrgyz),
*Min-em keçkene ul-um məktəp-qə bar-a* (Tatar).

But for the development of the MT system from Turkic languages into (for example) English (analytical language) or Russian (inflectional language) we have essentially different situation.

In these cases we should translate suffix chains in whole (without separating into simple suffixes) (Section 2) and this fact causes some problems. By adding various suffixes to the stem of the same verb, it is possible to create 17947 word-forms in Tatar [4], 11390 in Turkish and 13592 in Uzbek [5]. In the Kazakh language, the number of suffixes that create various word-forms from noun stems is about 500, while most verbs can be used up to 1000 various forms [6]. In Azerbaijani, the number of word-forms formed from the same stem is more than 8000 [7].

For these two reasons (the great number of suffix chains and the necessity to translate suffix chains in whole), it is necessary to define the subset of suffix chains used in texts i nstead of the set of every possible suffix chains.

Despite the great number of suffix chains we should also take into account that the frequency with which all these suffixes and their chains are used is not the same.

If this fact is considered while creating an MT system, then the diff iculties related to the great number of suffix

978-83-60810-14-9/08/$25.00 © 2008 IEEE

chains can be avoided. That's to say not all suffix chains, but the suffix chains used in written texts can be determined and included in the database of suffix chains.

Table A.
Turkic multilingual dictionary

| Azerb. | Turkish | Uzbek | Kazakh | Kyrgyz | Tatar |
|---|---|---|---|---|---|
| mən | ben | men | men | men | min |
| kiçik | küçük | kiçik | bəlekey | kiçine | keçkene |
| oğl | oğl | ўғл | bala | uul | ul |
| məktəb | okul | maktab | mektep | mektep | məktəp |
| ged | gid | bor | bar | bar | bar |

Table B.
Database of equivalency of simple suffixes

| | | | | | |
|---|---|---|---|---|---|
| im | im | ing | in | in | em |
| um | um | im | m | um | um |
| ə | a | qa | ke | ke | qe |
| ir | iyor | di | ır | dı | a |

Suffix chains regularly found in texts, we call *active* suffix chains and our purpose in this paper is to define the subset of active suffix chains of the Azerbaijani.

Hereinafter Azerbaijani is taken as a source and English as a target language and as a translation process, we will mean from Azerbaijani into the language not belonging to the Turkic group.

## II. Suffix Chains and Their Translations

In this section we will show the necessity of taking suffix chains in whole in the translation process.

As mentioned above, compound suffixes that consist of several simple suffixes are called *suffix chains* [8]. We will call the number of simple suffixes that comprise a suffix chain the *length* of this chain. Will also be referred simple suffixes as a suffix chain (whose length is one).

In the grammar of the modern Azerbaijani language, suffixes are divided into two groups—lexical and grammatical suffixes [8]. Lexical and grammatical suffixes form various word-forms from the same word stem by joining word stems both separately and in a certain sequence (for example, it is possible to form the word-forms *ev-də, ev-dəki-lər-in, ev-də-dir-lər, ev-dəki-lər-dən-siniz-mi* and etc. from the stem *ev* , Table 1 *)*.

While a word-form is translated into another language, it is necessary to take into account the meaning of each simple suffix that is included in the suffix chain.

At this moment, there is a question: is it possible to get the correct translation of a word-form by translating each simple suffix of its suffix chain separately and then by putting these translations together? It is very important question from the point of view of the development of the MT system from Azerbaijani, but as we will see below , the answer to this question is negative (Section 2).

Let's have a look at the examples shown below (suffixe chains and their translations are underlined in the same way) (Table 2). While the suffix - *də* is translated as *at* and the suffix - *dir* is translated as *he/she/it is* in the first example, in the second example, the suffix - *dir* cannot be translated separately. In this example, the suffix chain - *dir-lər* should be translated as *they are* . In the third example, the suffix chain —*dir-lər* is translated as *they are* and as the plural suffix—*s* which is added to the end of the word. In the fourth example, the rule of translating this chain changes again and the suffix chain - *dir-lər-mi* is translated as *are they* with the plural suffix - *s* added to the end of the word-form. The number of these examples can be increased.

Table 1.
Database of the active word-forms (fragment)

| Word-form | Word-form |
|---|---|
| *1.* ev | *14.* ev-indən |
| *2.* ev-dir | *15.* ev-inizə |
| *3.* ev-də | *16.* ev-inin |
| *4.* ev-dən | *17.* ev-inə |
| *5.* ev-i | *18.* ev-lə |
| *6.* ev-idir | *19.* ev-lər |
| *7.* ev-imdə | *20.* ev-lərdə |
| *8.* ev-imin | *21.* ev-lərdən |
| *9.* ev-imə | *22.* ev-ləri |
| *10.* ev-in | *23.* ev-lərin |
| *11.* ev-ində | *24.* ev-lərində |
| *12.* ev-lərinə | *25.* ev-lərindən |
| *13.* ev-indədir | … |

It is necessary to note, that these examples are right not only for English but also for other languages of non-Turkic group (Table C).

Table 2.
Examples to the translation of suffix chains into English

| Word-form | Stem of the word-form | Suffix chain | Translation of word-form |
|---|---|---|---|
| 1. *evdədir* | *ev*<br>home | *də* - *dir* | he is at home |
| 2. *evdədirlər* | *ev*<br>home | *də* - *dir-lər* | they are at home |
| 3. *tələbədirlər* | *tələbə*<br>student | *dir-lər* | they are student s |
| 4. *tələbədirlərmi* | *tələbə*<br>student | *dir-lər-mi* | a re they stu-dent s |

Table C.
Examples to the translation of suffix chains into Russian

| Azerbaijani su ffix | Russian equivalent |
|---|---|
| *-ir, - ır, -ur, -ür*<br>( *ged-ir* - he goes ) | *– ет , - ёт , - ит*<br>( *ид-ет* ) |
| *-lər, -lar*<br>( *kitab-lar* – books ) | *– и , - ы , - а , - я*<br>( *книг-и* ) |
| *-ir-lər*<br>( *ged-ir-lər* – they go ) | *-ут, -ют, -ат, -ят*<br>( *ид-ут* ) |

As it can be see n from the examples, the compositional translation of each suffix included in the suffix chain can lead to erroneous results. In order to get the right translation, the suffix chain should be translated as a whole, taking into account the meaning of each simple suffix which this chain is composed of.

So, for the development of an MT system, first we would define the set of not all, but only active suffix chains and to develop the translation rules of these chains.

## III. Active suffix Chains of Azerbaijani

In the previous section we have show n that in the translation process suffix chains must be taken in whole for correct translation of their meanings. In this section we will define the set of active suffix chains of the Azerbaijani. Creation of the suffix chains databases is also necessary for the morphological parsing. One of the main purposes of the formal morphological analysis in the Azerbaijani language is to separate the stem of the word-form from its suffix chain, because like all agglutinative languages, grammatical relations among word-forms in the Azerbaijani sentence are determined by the suffix chains and the correct determination of the suffix chain has a serious influence on the correct conduct of the further analysis process.

We should especially point out here that when we say suffixes and suffix chains, we only mean grammatical suffixes and suffix chains formed from them (Number of simple grammatical suffixes of Azerbaijani is about 100 [8]). We are not examining simple lexical suffixes or lexical suffixes in suffix chains, because words formed by lexical suffixes are kept as a separate lexical unit in the dictionary of MT system (this applies both to prefixes – *na, bi, ba, la, a, anti* and to other lexical suffixes - *lı, -li, -lu, -lü, -çı, -çi, -çu -çü, -lıq, -lik, -luq, -lük* etc.).

For example, although the words *balıq* (fish), *balıq-çı* (fisherman) and *balıq-çı-lıq* (fishery) come from the same stem, all three words are kept in dictionary as a separate lexical unit. Because, formally there are not general rules to generate the translation of the word-forms *balıq-çi* (fisherman), *araba-çı* (wagoner) etc. from the translation of the words *balıq* , *araba* (wagon) etc.

Before developing the MT system for the Azerbaijani language, the text corpus on various subjects with more than 300,000 word-forms was created (Later, volume of the text corpus has been increased to more than 12 million word-forms). Number of word-forms found in this corpus was about 39000 (after some processing). These word-forms were put in the database and stems of these word-forms were separated from their suffix chains manually as it presented in the Table 1.

During this process suffix chains were encountered 111,406 times, but the number of various suffix chains was 1,415 (with all variants of the suffix chains with the same meaning). These suffix chains form the basis of the "Database of suffix chains" of the Azerbaijani-English MT system. After grouping these chains (the ones that have the same function, but have different spelling, for example, *acaq, acağ, əcək, əcəy, yacaq, yacağ, yəcək, yəcəy),* the number of suffix chains reached 627.

The length of suffix chains was also calculated during the computer analysis. The arrangement of suffix chains by their length is given in the Table 3 (as we said above, when we talk about the length of suffix chains, we mean the number of simple grammatical suffixes that form this chain). As we can see from these results, very long suffix chains are rare and such chains almost are not used in writing. This can be clearly seen from Table 3. There are no suffix chains longer than five simple suffixes in the texts that were analyzed. The following table shows the frequency with which suffix chains are used by their length.

One of the possible reasons why suffix chains longer than five simple suffixes are not encountered could be that we did not take into account the lexical suffixes. On the other hand, the fact that long suffix chains are not used shows that although the use of such suffix chains is principally possible, no author uses them in writing or if it is necessary, the idea to be expressed by means of a long suffix chain is expressed by a shorter suffix chain (or words) that have the same meaning. For example: the sentence " *Siz bizim dəvət et-di-k-lər-imiz-dən-siniz-mi* " (Are you one of the people who we invited?) is replaced with an equivalent sentence " *Biz Sizi dəvət et-miş-ik-mi* " (Have we invited you?) or another similar equivalent sentence, for example, with the sentence " *Siz dəvət edil-mi-siniz-mi* " (Have you been invited?) A chain that has seven simple suffixes is replaced with a chain that has three simple suffixes.

As it can be seen from Table 3, a chain of five simple suffixes was encountered five times (0.004% of all cases), a chain of four simple suffixes was encountered 223 times (0.200%), a chain of three simple suffixes was encountered 6,895 times (6.189%), a chain of two simple suffixes was encountered 41,331 times (37.099%) and a chain of one suffix was encountered 62,952 times (56.507%).

The distribution of suffix chains that have the same functions without taking into account repetition was as follows (Table 3).

The number of various chains of five simple suffixes was four, the number of chains of four simple suffixes was 66, the number of chains of three simple suffixes was 248, and the number of chains of two simple suffixes was 257 while the number of chains of one simple suffix was 53.

TABLE 3.
FREQUENCY OF SUFFIX CHAINS

| **Length of chain** | **Frequency** | **Percentage of repeat** | **Unrepeated chains** |
|---:|---:|---:|---:|
| 5 | 5 | 0.004% | 4 |
| 4 | 223 | 0.200% | 66 |
| 3 | 6,895 | 6.189% | 248 |
| 2 | 41,330 | 37.099% | 256 |
| 1 | 62,952 | 56.507% | 53 |
| Total | 111,405 | 100.000% | 627 |

Based on these figures, we can say that most of all the suffix chains used in the Azerbaijani language are consisted of one, two or three simple suffixes.

These suffix chains com pound 99.795% of all most frequent suffix chains. Relative long suffix chains (the ones that have four, five simple suffixes and longer) compound only 0.205% of all chains.

The results that we obtained were analyzed again within the text corpus of 12 million word-forms. Although the volume of the text corpus is increased 40 times, the number of suffixes is increased 1.31 times, while the use of longer suffix chains did not change (that's to say an additional 196 suffix chains were determined and the number of encountered suffixes was 823). After getting this result it is possible to say confidently that active suffix chains of the Azerbaijani language do not exceed 1000.

The fact that the analyzed text corpus has a sufficiently great volume allows us to say that the expansion of the volume of the text corpus will not cause a considerable change in relative frequency indicators.

Besides, the types of active suffix chains on the definition of their capability to join the stems belonging to the different parts of speech are also determined. Because, besides well known ambiguity problems (lexical, syntactical etc.), there are grammatical ambiguity (ambiguity of suffixes) in agglutinative languages else and this information is used in the disambiguation process. 534 chains (≈64.88%) of all chains can join only verb stems (verb chain), 254 chains (≈30.86%) can join non-verb stems (non-verb chain) and 35 chains (≈4.25%) can join both types of stems (dual chain). For the dual chains their frequency of the using as verb or non-verb chains is also defined and this statistics is also used in lexical and grammatical disambiguation process.

So, we have defined the composition of the main information included in the database of the active suffix chains.

The fragmen t of this database is shown in the Table 4.

In the 3rd column of the table 4 are indicated the types of suffix chains. The letter "*V*" written in the third column shows that this suffix chain is a verb chain, while the letter "*N*" - non-verb chain. If none of these letters is written there, the suffix chain is a dual chain.

TABLE 4
DATABASE OF ACTIVE SUFFIX CHAINS (FRAGMENT)

| **Suffix chain** | **Other variants of suffix chains** | **Structure of the chain** | **Type** |
|---|---|---|---|
| *am* | *əm, yam, yəm* | | |
| *da* | *də* | | *N* |
| … | | | |
| *ı r* | *ur, ür, yur, yür, ir, yır, yir* | | *V* |
| *uram* | *ürəm, yuram, yürəm, ıram, irəm, yıram, yirəm* | *ur-am* | *V* |
| *da* | *də* | | *N* |
| *dadır* | *dədir* | *da-dır* | *N* |
| … | | | |
| *lar* | *lər* | | *N* |
| *larda* | *lərdə* | *lar-da* | *N* |
| … | | | |
| *mış* | *miş, muş, müş* | | |
| *mışdır* | *mişdir, muşdur, müşdür* | *mış-dır* | *V* |
| … | | | |

In addition, we would like to note that the database of the active suffix chains of the Dilmanc MT system has a more complex structure, but only necessary information used for morphological analysis in examples is presented here.

I n the next section we consider the use of the database of the active suffix chains in the morphological analysis process of the Azerbaijani word-forms.

## IV. THE USE OF THE ACTIVE SUFFIX CHAINS DATABASE IN MORPHOLOGICAL ANALYSIS PROCESS

The formation of the grammatical relations among word-forms in a sentence can appreciably differ for different languages. In analytical languages (for example: in English) the grammatical relations among word-forms in a sentence, in most cases, are defined by word order and/or prepositions. In analytical languages, separate words don't have grammatical information and such information can only be acquired in the existence of strict word order. But in agglutinative languages (for example, all the languages of the Turkic group are agglutinative), grammatical relations among word-forms in a sentence are formed by the rich set of suffix chains. For the definition of the grammatical relation among the word-forms at first it is necessary to separate stem and suffix chain of the word-form for the definition of the participation of the word-form in the syntactic structures of the sentence. For this reason morphological analysis algorithms are different for analytical and agglutinative languages.

In agglutinative languages, formal (by computer) morphological analysis can be carried out by creating *a Dictionary of stems* and *a Database of suffix chains*. The dictionary of the Azerbaijani word stems is also developed in frame of the project and in the Table 5 is indicated the simplified version of this dictionary (The dictionary of the Azerbaijani-English MT system has a more complex structure and most of information for the normal functioning of the translation algorithm is not presented here).

Not paying attention to the ambiguity problems, we will schematically describe the work of the morphological analyzer of Azerbaijani.

The morphological analysis algorithm of word-forms in Turkic languages is shown in [7]. This algorithm can be described shortly as follows:

1. The whole word-form is sought in the dictionary of stems (Table 5).
2. If the word-form is not found in the dictionary, its last letter is discarded and the remainder of the word-form – the truncated part is sought in the dictionary again. This process continues until the word-form or its truncated part is found in the dictionary of stems. Discarded part of the word-form is sought in the database of suffix chains (Table 4).
3. If discarded part of the word-form is a suffix chain and this chain can join the stem of this word-form (for example, if the stem is a verb, then the type of the suffix chain should be *V* – a verb chain), then this process stops, otherwise go to the second step;
4. After the stem and suffix chain of the word-form are defined, the word-form is provided with the information included in the databases of stems and suffix chains for its stem and suffix chain.

Example 1. Let's consider the formal morphological analysis process of the word-form *məktəbdədir* (he/she/it is in the school). Starting from the whole word-form all its reminders are sequentially sought in the dictionary of stems (Table 5). Only in the 6th step the stem *məktəb* of the word-form is found in the dictionary. Discarded part - *dədir* is also found in the database of suffix chains. So, process is stopped and we can right

*məktəbdədir* ⇔ *məktəb-dədir.*

The word-form and its remainders (according to above mentioned algorithm) with the discarding parts are shown below:

| | | |
|---|---|---|
| *1.* | *məktəbdədir* | |
| *2.* | *məktəbdədi* | *r,* |
| *3.* | *məktəbdəd* | *ir,* |
| *4.* | *məktəbdə* | *dir,* |
| *5.* | *məktəbd* | *ədir,* |
| *6.* | *məktəb* | *dədir*▲ |

The following examples show how to use the types of suffix chains in the formal morphological analysis process.

Example 2. Let's carry out a formal morphological analysis of the word-form *qorxuram* (I am afraid). According to the morphological analysis algorithm, in the 4th step – the word-form *qorxu* is sought and found in the Table 5. But discarded part of the word-form – *ram* is not suffix chain (Table 4). Therefore the process continues and in the 5th step – the word-form *qorx* (verb stem) is sought and found in the Table 5, discarded part – *uram* of this word-form is sought in the Table 4 and the process stops because such a suffix chain is found and it is verb chain.

Steps of this process are presented below:

1. *qorxuram*
2. *qorxura* *m,*
3. *qorxur* *am,*
4. *qorxu* *ram,*
5. *qorx* *uram.*

Thus, after this process we get

*qorxuram ⇔ qorx-uram* ▲

Example 3. For the word-form *yazır* (*yaz-ır,* he/she/it writes)

1. *yaz ır*
2. *yazı* *r,*
3. *yaz* *ır.*

in the 2nd step process does not stop, because *r* is not suffix chain (though *yazı* is found in the Table 5). In the next step two variants of the stem *yaz* (verb and noun) are found in the Table 5. Because discarded part – *ır* is a verb chain (Table 4) we chose the verb variant of the stem *yaz* ▲

Note that information included in the databases of stems and active suffix chains does not lead to the full solution of the lexical and grammatical ambiguity problem and it is necessary to return to the solution of the ambiguity problem in the next stages of the formal grammatical analysis (syntactic, semantic etc.).

Table 5.

Dictionary of Dilmanc MT system (Fragment)

| Stem | Part of speech | English translation |
|---|---|---|
| *tərcümə et* | verb | translate |
| … | | |
| dilmanc | noun | translator |
| … | | |
| *yaz* | verb | write |
| *yaz* | noun | spring |
| *yaz ı* | noun | record |
| … | | |
| *qur* | verb | construct |
| *quru* | verb | dry |
| *quru* | adverb | dry |
| *quru* | noun | land |
| … | | |
| *qorx* | verb | play |
| *qorxu* | noun | fear |
| … | | |
| *cədvəl* | noun | table |
| … | | |

## V. Dilmanc MT system

Despite some research works most of languages of Turkic group are still less investigated languages, except modern Turkish [9]-[13].

Researches on the development of Speech and NLP technologies for the Azerbaijani language are being led since 2003 [14]-[16]. Because Azerbaijani is one of less-investigated languages, the most of the necessary works (development of the MT dictionaries, creation of the formal grammar for Azerbaijani, algorithms for the automation of the translation process from/into Azerbaijani, synthesizer and analyzer algorithms of the Azerbaijani sentences, definition of the threephone set for the ASR system etc.) for the development of these technologies are carried out for the first time. The research work presented in this paper is one of such important steps on the creation of the applied linguistic technologies for Azerbaijani (All researches are carried out within the joint projects "Development of the MT system for Azerbaijani", "Development of the Speech Recognition system for Azerbaijani" of the Ministry of ICT of Azerbaijan and UNDP-Azerbaijan).

Dilmanc MT system is a hybrid MT system developed on the basis of RBMT (Rule Based MT) and SBMT ( Statistic Based MT) approaches.

Dilmanc MT system can translate for the present on three directions – Azerbaijani-English, English-Azerbaijani and Turkish-Azerbaijani (www.dilmanc.az). For the definition of the factors influencing the translation quality, first the set of test sentences consisting of 1000 sentences is formed. On the results of the test it is possible to say that the system gives good enough - intelligible translations in the most cases (http://www.science.az/cyber/pci2008/1.htm/1-26.doc).

Dilmanc MT system has the following characteristics on each direction (all these items have been developed for the first time):

*Azerbaijani-English direction.*

1. MT dictionary of Azerbaijani word stems (≈120000 lexical units including word phrases and terms);
2. Database of the active suffix chains (≈1000 active chains);
3. Database of the formalized rules for the decision of the lexical and syntactical ambiguity in Azerbaijani (≈1500 rules);
4. Database of translations of the active suffix chains of Azerbaijani (≈2300 rules);
5. Database of the formal signs of the parts of the sentence in Azerbaijani (≈2000 signs);
6. Formalized rules of the "traditional" grammar of Azerbaijani for the definition of the noun and verb phrases;
7. Formal morphological analysis algorithms of Azerbaijani word-forms;
8. Formal syntactic analysis algorithms of the Azerbaijani sentences;
9. Algorithms for the synthesis of the English sentences.

*English-Azerbaijani direction.*

1. English-Azerbaijani MT dictionary (≈115000 lexical units including word phrases and terms);
2. Database of the formalized rules for the decision of the lexical and syntactical ambiguity (≈1400 rules);
3. Database of the formalized rules for the synthesis of Azerbaijani suffix chains (≈300 rules);
4. Database of the rules for delimitation of the homogeneous parts in the English sentence (≈90 rules);
5. Database of the rules for delimitation of clauses in the English sentence (≈40 rules);
6. Algorithms for the formal syntactic analysis of the English sentences.
7. Algorithms for the synthesis of the Azerbaijani sentences.

*Turkish-Azerbaijani direction.*

1. Turkish-Azerbaijani MT dictionary (≈20000 lexical units);
2. Database of the equivalency of Turkish and Azerbaijani suffix chains (≈1000 chains).

It is necessary to note that this list is only a small part of all algorithmic and non-algorithmic means developed in the frame of the Dilmanc MT system.

In addition the formed set of active suffix chains may be used while developing other linguistic technologies as speech and other NLP systems.

Although the analyses are carried out for the Azerbai jani language, there is no doubt that this approach is also applicable for other Turkic languages.

## References

[1] Altıntaş K., Çiçekli İ. "A Morphological Analyzer for Crimean Tatar." *In: Proceedings of the 10th Turkish Sy mposium on Artificial Intelligence and Neural Networks,* TAINN, pp. 180-189, North Cyprus.

[2] Cüneyd Tantuğ, Eşref Adalı and Kemal Oflazer, "A MT System from Turkmen to Turkish Employing Finite State and Statistical Methods," in *Proceedings of MT Summit XI,* 2007.

[3] Cüneyd Tantuğ, Eşref Adalı and Kemal Oflazer, "Machine Translation between Turkic Languages," in *Proceedings of ACL 2007–Companion Volume,* Prague, Czech Republic, June 2007.

[4] Iskhakova Kh.F (1968) "Avtomaticheskiy sintez form sushestvitelnogo v tatarskom yazike." *Sovetskaya tyurkologiya,* 2(8): 20-27.

[5] Pines V. Y. (1974) "Nekotorie voprosi avtomaticheskogo perevoda i tyurkskie yaziki." *Sovetskaya tyurkologiya,* 3:100-107.

[6] Bektayev K. (1990) Statistika kazakhskogo teksta. Gilim, Almaati.

[7] Mahmudov M. (2002) Metnlerin formal tehlili sistemi. Elm, Baku.

[8] Abdullayev A., Seyidov Y., Hasanov A. (1972) Müasir Azərbaycan dili (Modern Azerbaijani language). Maarif, Baku.

[9] Cicekli I., Korkmaz T. (1998) "Generation of Simple Turkish Sentences with Systemic-Functional Grammar." In: *Proceedings of the 3rd International Conference on New Methods in Language Processing (NeMLaP-3),* Sydney, Australia, January 1998.

[10] Durgar-El-Kahlout I., Oflazer K. (2006) "Initial Explorations in English to Turkish Statistical Machine Translation." *Workshop on Statistical Machine Translation,* New York, NY, June 2006.

[11] Temizsoy M., Cicekli I. (1998) "An Ontology-Based Approach to Parsing Turkish Sentences." In: *Proceedings of AMTA'98-Conference of the Association for Machine Translation in the Americas,* Lecture Notes in Computer Science 1529, Springer Verlag, October 1998, Langhorne, PA, USA.

[12] Tur G., Hakkani-Tur D., Oflazer K. (2000) " Statistical Modeling of Turkish for Automatic Topic Segmentation ." *Bilkent University, Computer Engineering Technical Report BU-CE-0001,* January 2000.

[13] Vural E., Erdogan H., Oflazer K., Yanikoglu B. (2005) "An Online Handwriting Recognition System For Turkish." In: *Proceedings of SPIE* Vol. 5676 Electronic Imaging 2005, San Jose, January 2005.

[14] Abbasov A., Fatullayev A. (2007) "The use of syntactical and semantic valences of the verb for formal delimitation of verb word phrases." In: *Proceedings of the 3rd La nguage & Technology Conference (L&TC'07).* 5-7 October 2007, Poznan, Poland.

[15] Fatullayev A., Mehtaliyev A., Ahmedov F., Fatullayev R. (2004) "Computer translation system from Azerbaijan language into English." *Proc. of the 4th international conference Internet-Education-Science,* Vinnitsia, 2004, vol. 2, p. 572.

[16] Abbasov A. M., Fatullayev R. A. (2006) "English-Azerbaijani machine translation system on the basis of compressed templates and formal grammatical analysis." *Problems of cybernetics and informatics International Conference (PCI 2006).* Baku – 2006, pp. 42-45.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 369 – 373

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A New Word Sense Similarity Measure in WordNet

Ali Sebti
Amirkabir university of technology, Intelligence Systems Laboratory[1], Tehran, Iran
Email: ali.sebti@aut.ac.ir

Ahmad Abodollahzadeh Barfroush
Amirkabir university of technology
Intelligence Systems Laboratory
Tehran, Iran
Email: ahmad@ce.aut.ac.ir

*Abstract*—**Recognizing similarities between words is a basic element of computational linguistics and artificial intelligence applications. This paper presents a new approach for measuring semantic similarity between words via concepts. Our proposed measure is a hybrid system based on using a new Information content metric and edge counting-based tuning function. In proposed system, hierarchical structure is used to present information content instead of text corpus and our result will be improved by edge counting-based tuning function. The result of the system is evaluated against human similarity ratings demonstration and shows significant improvement in compare with traditional similarity measures.**

## I. Introduction

SEMANTIC similarity is an important topic in natural language processing (NLP) and Information Retrieval (IR). It has also been subject to studies in Cognitive Science and Artificial Intelligence. Application areas of semantic similarity include word sense disambiguation (WSD) [19], information extraction and retrieval [2,22,24], detection and correction of word spelling errors (malapropisms)[3], text segmentation [10], image retrieval [21], multimodal document retrieval [20], and automatic hypertext linking [5], automatic indexing, text annotation and summarization [13].

To quantify the concept of similarity between words, some ideas have been put forth by researchers, most of which rely heavily on the knowledge available in lexical knowledge bases like WordNet.

There are mainly two approaches to compute semantic similarity. The first approach is making use of a large corpus or word definitions and gathering statistical data from these sources to estimate a score of semantic similarity, which we call text-based approach. The second approach makes use of the relations and the hierarchy of a thesaurus, such as WordNet, which we call structure-based approach.

In text-based approach, word relationships are often derived from their co-occurrence distribution in a corpus [7,6]. Gloss overlap, introduced by Lesk [12] and extended gloss overlap, introduced by Banerjee and Pedersen, are another instances of this approach. The latter is a measure that determines the relatedness of concepts proportional to the extent of overlap of their WordNet glosses [1]. Besides gloss vector measure of semantic relatedness, introduced by Pedersen and Patwardhan, is based on second order co–occurrence vectors in combination with the structure and content of WordNet, a semantic network of concepts [16].

In structure-based approach, first studies date back to Quilian's semantic memory model [17], where the number of hops between nodes of concepts in the hierarchical network specifies the similarity or difference of concepts. Wu and Palmer's semantic similarity measure was based on the path length between concepts located in a taxonomy [23]. Also, the similarity measure of Leacock and Chodorow is based on the shortest path length between two concepts in is-a hierarchy [11].

In combining two approaches, Resnik introduced a new factor of relatedness called information content (IC) [18]. The Similarity measures of Resnik, Jiang and Conrath [9] and Lin [14] all rely on the IC values assigned to the concepts in an is-a hierarchy, but their usage of IC has little differences. Using a different approach Hirst G. and St-Onge assign relatedness scores to words rather than word senses. They set different weights for different kinds of links in a semantic network, and uses those weights for edge counting [8].

In this paper, we first introduce a new method for computing IC of concepts in a hierarchical structure. We will show that this method only uses hierarchical structure and not corpus to determine IC. Furthermore, information content obtained from this method implicitly includes depth and branch factor of the concept from root to target concept. Then we use formula that is similar to Lin formula for measuring similarity. Then we analyze our result and comparing it with benchmark result and introduce an edge counting-based function for improving and overcome their problems. For adjusting our function's parameters we use genetic algorithm. Finally our combined similarity measure is evaluated against a benchmark set of human similarity ratings, and demonstrates that the proposed measure significantly outperformed traditional similarity measures.

In section 2 we describe WordNet, which was used in developing our method. Section 3 describes the extraction of our new information content metric from a lexical knowledge base. Section 4 presents the choice and organization of a benchmark data set for evaluating the similarity method, how to define a tuning function, experimental results and dis cussion about it. Finally, paper concludes in Section 5 that,

[*] Authors thanks Iran Telecommunication Research Center (ITRC) for their support during this research work

[1] http://ceit.aut.ac.ir/islab/

based on the benchmark data set, our measure outperforms existing measures.

## II. WordNet

WordNet is the product of a research project at Princeton University which has attempted to model the lexical knowledge of a native speaker of English [4]. In WordNet each unique meaning of a word is represented by a synonym set or *synset*. Each synset has a gloss that defines the concept of the word. For example the words *car*, *auto*, *automobile*, and *motorcar* is a synset that represents the concept define by gloss: *four wheel Motor vehicle, usually propelled by an internal combustion Engine*. Many glosses have *examples* of usages associated with them, such as *"he needs a car to get to work."*

In addition to providing these groups of synonyms to represent a concept, WordNet connects concepts via a variety of semantic relations. These semantic relations for nouns include:

- Hyponym/Hypernym (IS-A/ HAS A)
- Meronym/Holonym (Part-of / Has-Part)
- Meronym/Holonym (Member-of / Has-Member),
- Meronym/Holonym (Substance-of / Has-Substance)

Figure 1 shows a fragment of WordNet taxonomy.

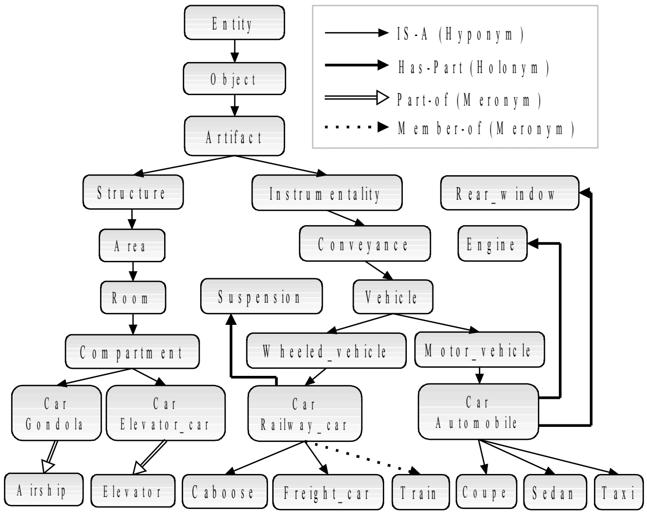


Fig. 1 fragment of WordNet taxonomy

## III. The new Information Content Metric

### A. Previous information content based approaches

Many researchers consider statistical figures to compute IC value. They assign a probability to a concept in taxonomy based on the occurrence of target concept in a given corpus. The IC value is then calculated by negative log likelihood formula as follow:

$$IC(c)=-\log(p(c)) \qquad (1)$$

Where c is a concept and p is the probability of encountering c in a given corpus. Philip Resnik [18] used this formula to compute semantic similarity between concepts. Basic idea behind the negative likelihood formula is that the more probable a concept appears, the less information it conveys, in other words, infrequent words are more informative then frequent ones.

Resnik showed that semantic similarity depends on the amount of information that two concepts have in common, this shared information is given by the Most Specific Common Abstraction (MSCA) that subsumes both concepts. Therefore we must first discover the MSCA and then shared information is equal to the IC value of the MSCA. If MSCA does not exist then the two concepts are maximally dissimilar. Formally, Resnik semantic similarity is defined as:

$$sim_{res}(c_1,c_2)=\max_{c\in S(c_1,c_2)} ic_{res}(c) \qquad (2)$$

where *S(c₁,c₂)* is the set of concepts that subsume $c_1$ and $c_2$. Another information theoretic similarity metric that used the same notion of IC was that of Lin [23], expressed by:

$$sim_{lin}(c_1,c_2)=\frac{2\times sim_{res}(c_1,c_2)}{(ic_{res}(c_1)+ic_{res}(c_2))} \qquad (3)$$

Jiang and Conrath [9] also proposed a new measure of semantic distance that its corresponding semantic similarity can be obtained from the reverse of it. Common version of their distance metric is:

$$\begin{aligned} dist_{jcn}(c_1,c_2)=&(ic_{res}(c_1)+ic_{res}(c_2)) \\ &-2\times sim_{res}(c_1,c_2) \end{aligned} \qquad (4)$$

### B. Our new information content metric

Our method of obtaining IC values is based on the assumption that the taxonomic structure of WordNet is organized in a meaningful and principled way, where concepts in higher depths and having more sibling concepts in the taxonomy structure are more informative and their IC values are bigger. Our method includes implicitly these two parameters that figure 2 represent this method for computing IC value for a fragment of concepts in WordNet.

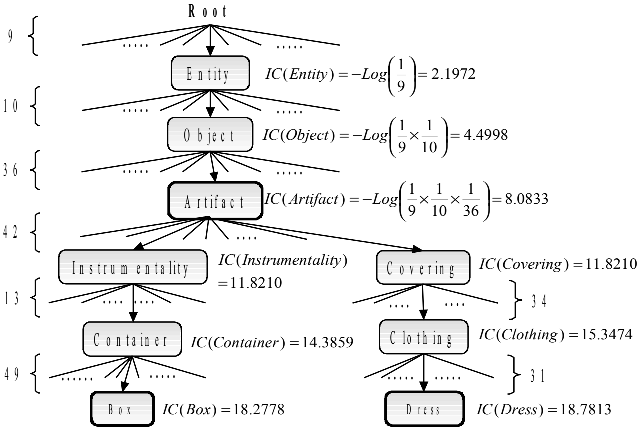

Root
9
Entity
$IC(Entity)=-Log\left(\frac{1}{9}\right)=2.1972$
10
Object
$IC(Object)=-Log\left(\frac{1}{9}\times\frac{1}{10}\right)=4.4998$
36
Artifact
$IC(Artifact)=-Log\left(\frac{1}{9}\times\frac{1}{10}\times\frac{1}{36}\right)=8.0833$
42
Instrumentality
$IC(Instrumentality)=11.8210$
Covering
$IC(Covering)=11.8210$
13
34
Container
$IC(Container)=14.3859$
Clothing
$IC(Clothing)=15.3474$
49
31
Box
$IC(Box)=18.2778$
Dress
$IC(Dress)=18.7813$

Fig. 2 example of computing our IC metric for some concepts

For better understanding of this method we show in equation 5 how IC value of *Box* is computed according to figure 2:

$$IC(Box) = -Log\left(\frac{1}{9}\times\frac{1}{10}\times\frac{1}{36}\times\frac{1}{42}\times\frac{1}{13}\times\frac{1}{49}\right) = 18.2778 \qquad (5)$$

## IV. Implementaion

### A. Semantic similarity measure

To evaluate the effect of our Information Content Metric on semantic similarity, we first select an existing semantic similarity measure. For this purpose we use Lin semantic similarity measure. This approach makes the implementation easier with less complexity. Lin's formula is shown in equation 3.

### B. Benchmark data

In accordance with previous research, we evaluated the results by correlating our similarity scores with that of human judgments provided by Miller and Charles [15]. In their study, 38 undergraduate subjects were given 30 pairs of nouns and were asked to rate similarity of meaning for each pair on a scale from 0 (no similarity) to 4 (perfect synonymy). The average rating for each pair represents a good estimate of how similar the two words are. This benchmark data is used by many researchers in semantic similarity subject [1,16].

### C. Edge counting-based tuning function

For beginning our analysis, we first compute semantic similarity between pairs of words with Lin similarity and our similarity approach. As said before, our semantic similarity formula is the same as Lin formula. The difference between them is the method of computing IC value . Then, we draw our obtained result and Lin result and human judgments scores in a diagram. These results are showed in figure 3. As shown in figure 3, in some pairs of words our method is more accurate and in some pairs Lin similarity measure is better. We then decide to improve our accuracy in pairs that our method is less accurate. Therefore, the Next step is how we determine these pairs of words automatically. In others words we must define a new feature for pairs of words that discriminate pairs of words that our similarity method about them is less accurate toward Lin method.

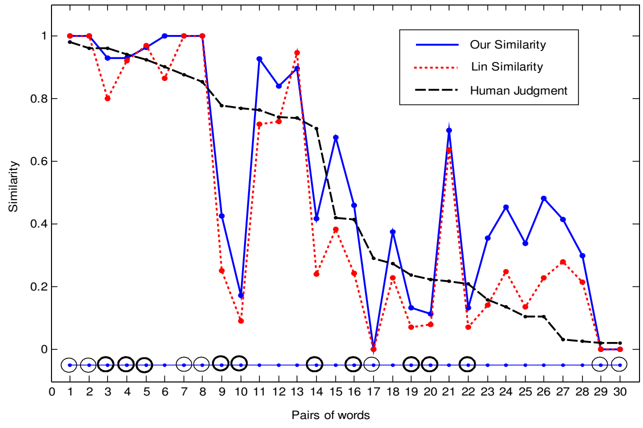


Fig 3 Compare our method with Lin and Human judgments

In figure 3 we show two types of circles: bold line circle and normal line circle. Bold line circles represent pairs of words that accuracy of our method is better than Lin and normal line circle shows that Lin and our method are the same. For other pairs, Lin method is more accurate. As said before in this step we extract a feature that determines inaccurate pairs. Table 1 shows our result, Human judgments, path of two words (concept) and depth of Lowest Common Subsumer (LCS) for two words. In this table if does not exist LCS for a pair, values of LCS depth and Path are -1.

Table 1
Result of our method, human judgment and three features

| Pairs of words | HJ | Ours | LCS depth | Path | $(LCS_{depth}+1)$ /(path+1) |
|---|---|---|---|---|---|
| car -automobile | 0.98 | 1 | 8 | 0 | 9 |
| gem – jewel | 0.96 | 1 | 6 | 0 | 7 |
| Journey - voyage | 0.96 | 0.93 | 5 | 1 | 3 |
| boy – lad | 0.94 | 0.93 | 4 | 1 | 2.5 |
| coast – shore | 0.92 | 0.96 | 4 | 1 | 2.5 |
| asylum -madhouse | 0.90 | 1 | 7 | 1 | 4 |
| magician – wizard | 0.87 | 1 | 4 | 0 | 5 |
| midday - noon | 0.85 | 1 | 7 | 0 | 8 |
| furnace - stove | 0.77 | 0.42 | 2 | 10 | 0.27 |
| food – fruit | 0.77 | 0.17 | 0 | 7 | 0.12 |
| bird - cock | 0.76 | 0.92 | 7 | 1 | 10 |
| bird - crane | 0.74 | 0.84 | 7 | 3 | 2 |
| tool - implement | 0.73 | 0.89 | 4 | 1 | 2.5 |
| brother -monk | 0.70 | 0.41 | 2 | 5 | 0.5 |
| crane - implement | 0.42 | 0.67 | 3 | 4 | 0.8 |
| lad - brother | 0.41 | 0.46 | 2 | 4 | 0.6 |
| journey - car | 0.29 | 0 | -1 | -1 | 10 |
| monk - oracle | 0.27 | 0.37 | 2 | 7 | 0.37 |
| cemetery - woodland | 0.23 | 0.13 | 0 | 9 | 0.1 |
| food - rooster | 0.22 | 0.11 | 0 | 13 | 0.07 |
| coast - hill | 0.21 | 0.69 | 3 | 4 | 0.8 |
| forest - graveyard | 0.21 | 0.13 | 0 | 9 | 0.1 |
| shore - woodland | 0.15 | 0.35 | 1 | 5 | 0.33 |
| monk - slave | 0.13 | 0.45 | 2 | 4 | 0.6 |
| coast - forest | 0.10 | 0.34 | 1 | 6 | 0.28 |
| lad - wizard | 0.10 | 0.48 | 2 | 4 | 0.6 |
| chord - smile | 0.03 | 0.41 | 3 | 10 | 0.36 |
| glass - magician | 0.02 | 0.29 | 1 | 7 | 0.25 |
| noon - string | 0.02 | 0 | -1 | -1 | 10 |
| rooster - voyage | 0.02 | 0 | -1 | -1 | 10 |

Our result shows the pairs of words that their LCS depth is little or path length of two concepts is large, our result is less accurate. These two conditions are combined with new feature that can be seen in the sixth column in table 1. Therefore if the new feature is low, pairs of words which our result is related to, is less accurate and hence is detectable. In figure 4 we show that, in new feature space a specific bund contains most inaccurate pairs of words.

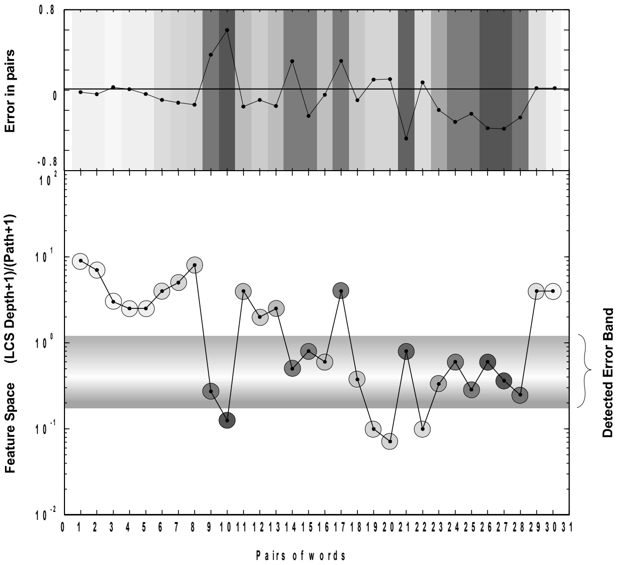


Fig 4 detecting error bund in new feature space

One considerable point is that, when new feature is too low, our similarity result is lower than human judgment and when not too low, our similarity result is higher than human judgment. This point persuades us to define a tuning function which modifies our result. Thus we define a function that its general shape is showed in figure 4. In equation 6 we show mathematical formula of this function.

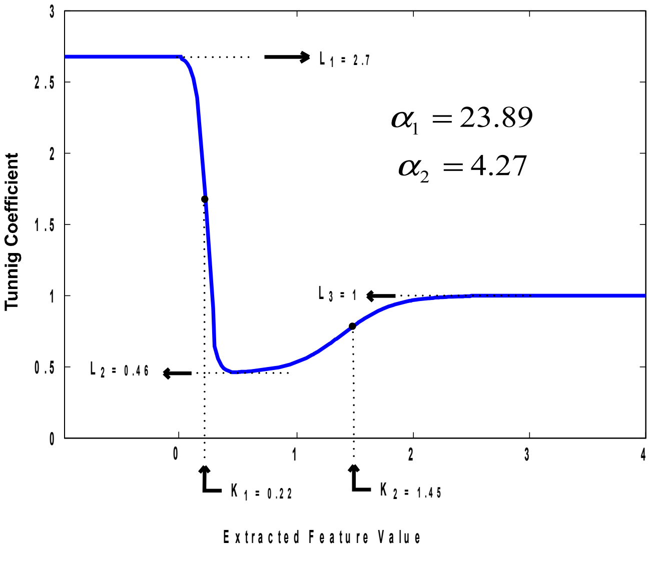


Fig 5 General shape of our tuning function

To determine the best value for parameters of this function we use genetic algorithm. Because of lack in data, for tuning parameters, we use Leave-one-out cross validation and then values of each parameters is average of the obtained values in 30 experiments. In Leave-one-out cross validation, for a dataset with N examples, perform N experiments. For each experiment use N-1 examples for training and the remaining example for testing. Figure 5 present the best value obtained by GA algorithm for $l_1$, $l_2$, $l_3$, $k_1$, $k_2$, $\alpha_1$ and $\alpha_2$. In table 2 we compare our final result with other methods. In all other similarity measures that use IC value, IC value is computed like Resnik's manner which was discussed in section III. For all of these experiments, Miller's benchmark data (30 pairs of words) is used. In order to make fair comparisons, we decided to use an independent software package that would compute similarity values using previously established strategies while allowing the use of WordNet 2.0. One freely available package is that of Siddharth Patwardhan and Ted Pederson [25]. This result shows that our similarity measure is comparable with other similarity measures.

$$f(x)=\begin{cases} x\le k_1-\frac{5}{\alpha_1} & l_1 \\ x>k_1-\frac{5}{\alpha_1} \quad and \quad x<k_1+\frac{5}{\alpha_1} & \frac{l_1-l_2}{1+\exp^{-\alpha_1(-x+k_1)}}+l_2 \\ x\ge k_1+\frac{5}{\alpha_1} \quad and \quad x\le k_2-\frac{5}{\alpha_2} & l_2 \\ x>k_2-\frac{5}{\alpha_2} \quad and \quad x<k_2+\frac{5}{\alpha_2} & \frac{l_3-l_2}{1+\exp^{-\alpha_2(x-k_2)}}+l_2 \\ x\ge k_1+\frac{5}{\alpha_2} & l_3 \end{cases} \tag{6}$$

TABLE 2
COMPARE OUR METHOD WITH OTHERS RELATED WORK IN CORRELATION WITH HUMAN JUDGMENT

| Similarity measure | correlation |
|---|---|
| Jiang and Conrath | 0.695 |
| Hirst St.Onge | 0.689 |
| Leacock Chodorow | 0.821 |
| Lin | 0.823 |
| Resnik | 0.775 |
| Wu and Palmer | 0.803 |
| Patwardhan and Pedersen | 0.77 |
| **Our Similarity Measure** | **0.87** |

## V. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a new word sense similarity measure with a proper tuning function. For computing information content, we used hierarchical structure alone, instead of text corpus. Experimental evaluation against a benchmark set of human similarity ratings demonstrated that the proposed measure significantly outperformed traditional similarity measures. In future work, we intend to that use this similarity measure in real world applications such as word sense disambiguation. Also, our tuning function can be used with other previous similarity measures.

## REFERENCES

[1] S. Banerjee and T. Pedersen. "Extended gloss overlaps as a measure of semantic relatedness". In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* , pages 805–810, Acapulco, Mexico, 2003.

[2] C. Buckley, J. Salton, J. Allen and A. Singhal, A. "Automatic query expansion using Smart: TREC 3". In *The third Text Retrieval Conference* , Gaithersburg, MD, 1995.

[3] A. Budanitsky and G. Hirst, "Semantic Distance in WordNet: An Experimental, Application-Oriented Evaluation of Five Measures". Proc. *Workshop WordNet and Other Lexical Resources, Second Meeting North Am. Chapter Assoc. for Computational Linguistics* , June 2001.

[4] C. Fellbaum, editor. "WordNet: An Electronic Lexical Database". *MIT Press*, Cambridge, USA, 1998.

[5] S. J. Green, "Building Hypertext Links by Computing Semantic Similarity". *IEEE Trans. Knowledge and Data Eng* , vol. 11, no. 5, pp. 713-730, Sept./Oct. 1999.

[6] G. Grefenstette. "Use of Syntactic Context to Produce Term Association Lists for Text Retrieval". *Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval* , SIGIR'92, 1992.

[7] D. Hindle. "Noun Classification from Predicate-Argument Structures". *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics* , ACL28'90, 268-275, 1990.

[8] G. Hirst and D. St-Onge. "Lexical chains as representations of context for the detection and correction of malapropisms". In *Fellbaum* , pp. 305–332, 1998.

[9] J. Jiang and D. Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy". In *Proceedings of International Conference on Research in Computational Linguistics* , Taiwan, 1997.

[10] H. Kozima, "Computing Lexical Cohesion as a Tool for Text Analysis". *doctoral thesis, Computer Science and Information Math* , Graduate School of Electro-Comm., Univ. of Electro-Comm., 1994.

[11] C. Leacock and M. Chodorow. "Combining local context and WordNet similarity for word sense identification". *In Fellbaum*, pp. 265–283, 1998.

[12] M. Lesk. "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone". In *Proceedings of the SIGDOC Conference,* Toronto, 1986.

[13] C. Y. Lin, and E. Hovy. "Automatic evaluation of summaries using n-gram co-occurrence statistics". In *Proceedings of Human Language Technology Conference (HLT-NAACL)* , Edmonton, Canada, 2003.

[14] D Lin. "An information-theoretic definition of similarity". In *Proceedings of the 15th International Conference on Machine Learning* , Madison, WI, 1998.

[15] G. Miller and W. Charles. "Contextual correlates of semantic Similarity". *Language and Cognitive Processes* , 6, 1–28, 1991.

[16] S. Patwardhan and T. Pedersen. "Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts". In *Proceedings of Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together* , EACL, .2006.

[17] M. R. Quilian. "Semantic memory". *Semantic Information Processing* . pages 216–270, 1968.

[18] P. Resnik. "Using information content to evaluate semantic similarity". In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* , pages 448–453, Montreal, 1995.

[19] P. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language". *J. Artificial Intelligence Research*, vol. 11, pp. 95-130, 1999.

[20] R. K. Srihari, Z.F. Zhang, and A.B. Rao, "Intelligent Indexing and Semantic Retrieval of Multimodal Documents". *Information Retrieval* , vol. 2, pp. 245-275, 2000.

[21] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years". IEEE *Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349-1380, Dec. 2000.

[22] O. Vechtomova and S. Robertson. "Integration of collocation statistics into the probabilistic retrieval model". In *22 nd Annual Colloquium on Information Retrieval Research* , Cambridge, England, , 2000.

[23] Z. Wu and M. Palmer. "Verb semantics and lexical selection". In *32nd. Annual Meeting of the Association for Computational Linguistics* . pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

[24] J. Xu, and B. Croft. "Improving the effectiveness of information retrieval". *ACM Transactions on Information Systems* , 18(1):79-112, 2000.

[25] http://wn-similarity.sourceforge.net

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 375–379

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Linguistic Light Approach to Multilingualism in Lexical Layers for Ontologies

Alexander Troussov, John Judge, Mikhail Sogrin, Amine Akrout
IBM
Email: {atrosso, johnjudge, sogrimik, amine_akrout}@ie.ibm.com

Brian Davis, Siegfried Handschuh
DERI
Email: {brian.davis, siegfried.handschuh}@deri.org

***Abstract*—Semantic web ontologies are being increasingly used in modern text analytics applications and Ontology-Based Information Extraction(OBIE) as a means to provide a semantic backbone either for modelling the internal conceptual data structures of the Text Analytics(TA) engine or to model the Knowledge base to drive the analysis of unstructured information in raw text and subsequent Knowledge acquisition and population. Creating and targeting Language Resources(LR)s from a TA to an Ontology can be time consuming and costly. In [1] the authors describe a user-friendly method for Ontology engineers to augment an ontologies with a lexical layer which provides a flexible framework to identify term mentions of ontology concepts in raw text. In this paper we explore multilinguality in these lexical layers using the same framework. We discuss a number of potential issues for the "linguistic light" Lexical Extensions for Ontologies (LEON) approach when looking at languages more morphologically rich and which have more complex linguistic constraints than English. We show how the LEON approach can cope with these phenomena once the morphological normaliser used in the lexical analysis process is able to generalise sufficiently well for the language concerned.**

## I. Introduction

SEMANTIC web ontologies are being increasingly used in modern text analytics applications and Ontology-Based Information Extraction(OBIE) as a means to provide a semantic backbone either for modelling the internal conceptual data structures of the Text Analytics(TA) engine or to model the Knowledge base to drive the analysis of unstructured information in raw text and subsequent Knowledge acquisition and population. Creating and targeting Language Resources(LR)s from a TA to an Ontology can be time consuming and costly. A Language Engineer working with a TA system must typically manually align existing internal linguistic resources with a new Ontology or create new LR's to support a domain shift. If the creation of LRs for an TA system is integrated into the Ontology engineering process via user fiendly Ontology lexicalisation for non-linguists. A lexical layer, which describes the various lexical realisations of Ontological term facilitates such a process. The "linguistic light" approach described in [1] outlines such a lightweight lexical layer which can be easily implemented into an existing ontology. The lexical layer (LEON)—(Lexical Extensions for Ontologies) can be subsequently traversed and compiled into internal LRs of the TA engine. Additionally, Organisations working in multilingual enviroments creates a demand for multilingual Ontologies [2] [1] The authors also claim that their approach can be retargetted to a new domain or language by simply providing the appropriate lexical information. The cross language portability of this approach and associated issues will be presented in this paper. Portability across languages is an important characteristic for an approach to lexical layers because of the cost and effort involved in redeveloping an ontology for a new language. One of the main principles behind the semantic web is the ability to easily exchange and utilise semantic information so by having a unified approach to identifying occurrences of Ontological terms in text across a number of languages we can maintain this inter-operability by using the same ontology.

This rest of this paper is structured as follows: Section II discusses related work, Section III gives an overview of LEON type lexical layers, Section IV describes how LEON can provide a multilingual lexical layer and highlights some potentially problematic features of languages besides English, Section VI explains how the LEON approach copes with these phenomena and discusses the implications of false positives, finally, Section VII conludes.

## II. Related Work

The inclusion of a linguistic or lexical layer into an Ontology or Ontology lexicalization is by no means a new phenomenon. For example, Linginfo, was developed as part of the SmartWeb[3] project[3]. The work conceptualized the idea of a linguistic layer for a Semantic Web Ontology or more specifically a "multilingual/multimedia lexicon model for Ontologies" [3]. Linguistic representation in LingInfo can consist of: a Language identifier, POS (Part of Speech) tag, morphological data, and syntactic compositional data as well a contextual data in the form of grammar rules of N-grams. Furthermore, content and knowledge are organized into four layers, where the Ontology layer is located at the central layer and linguistic features and their subsequent associations to the central layer are located in the outer middle layers with the outer layer containing textual content. The LingInfo model is applied to the SmartWeb Integrated Ontology

[1] Although against good Ontology Engineering practice, a substantial amount of Ontologies on the Web are in English which forces the need for localising knowledge. One can observe this easily by accessing such tools as OntoSelect[2]

[3] http://smartweb.dfki.de/

978-83-60810-14-9/08/$25.00 © 2008 IEEE

SWInto, whereby the linguistic feature layer is compiled into Language Resources (gazetteers) within the SProuT IE engine based on a mapping between the SWIntO and SProuTs TDL Type Description Languages. This mapping is applied to both SWIntO concepts and properties. The work of [3] is influenced strongly by LMF Lexical Markup Framework,[4], which is part of the ISO TC37/SC4[4] working group on the management of Language Resources. LMF has its origins in Language Engineering standardization initiatives such as EAGLES[5] and ISLE[6] . LingInfo also caters for multilingual Ontology lexicalisation, but we argue that the LingInfo model is too complex for use but non-linguistic engineers where LEON attempts to shield the Knowledge Engineer from complex linguistic formalisms.

Ontology lexicalisation is closely related to work within Lingusitic Ontologies. Linguistic Ontologies are used to describe semantic constructs rather than to model a specific domain and they are typically characterised by being bound to the semantics of grammatical or linguistic units i.e. GUM and SENSUS [5]. Ontologies such as Wordnet [6] and EuroWordnet [7] however are concerned with word meaning. Certain linguistic Ontologies are language independent such as EuroWordnet while the majority are not. EuroWordnet is a multilingual database containing wordnets for several Eurpean languages [8]. Each language specific word net is similarly structured to the English WorldNet and are linked via an Interlingua index. Consequently, one can access the translation of similar words in a target language for a given word within the source language. Linguistic Ontologies are primarily descriptive though they are frequently exploited by NLP systems either directly or to bootstrap the creation of new Language Resources. LEON on the other hand is designed explicitly to support the text analytics (or IE) task by replacing the manual retargeting of multilingual LRs within an IE system to an Ontology either (semi-)automatically.

Ontology Localization is also a closely related field to that of Ontology lexicalization. ”Ontology Localization consists of dapting an Otology to a concrete language and cultural community”[2]. In [2] the authors describe LabelTranslator, an Ontology localization tool which automatically translates ontological term labels (rdfs:labels of classes, instances and properties) in a source language to their target language equivalent. The system caters for English, German and Spanish. LabelTranslator attempts to best the most approximate translation by accessing translation services such as Babelfish and FreeTranslation, in addition to various Language Resources such as EuroWordnet [7], Wikitionary and GoogleTranslate. A ranking method based on the Normalized Google Distance(NGD) [9] is also applied to propose the most approximate target translation label from collection of suggested translations by taking into account the similarity of the source language label’s lexical and semantic context. The LEON approach is tailored towards an IE task which is very different from that of localisation, since as already shown in [1], rdfs:labels is a form of Ontology lexicalistion are too simplistic to capture the lingusitic idiosyncracies of certain surface forms as is the case with Multi Word Expressions.

Finally, we note other OBIE systems such as GATE[7] which can be deployed as a mulitilingual OBIE platform [10], however LRs in GATE must be manually aligned to the Ontology, while the LEON approach attempt to subsume part of the Dictionary creation process within the Ontology Engineering process.

## III. LEON

A lexical layer which describes the lexical realisations corresponding to concepts encoded in an ontology provides an interface between the ontology and text processing applications which seek to exploit the semantics encoded in the ontology.

The number and type of lexical expressions which correspond to a particular semantic entity varies from concept to concept, however, often they occur in a form which is different from the citation form because of inflectional or grammatical needs imposed by the language. These lexical realisations are often complex and appear as multiple word units, which in turn are not always fixed expressions and can vary depending on the context.

It would appear that an adequate approach to provide such a lexical layer requires some level of linguistic knowledge to be encoded alongside the semantics. This approach however becomes somewhat untenable in practice as there are many different linguistic theories to choose from which can lead to incompatibilities between ontologies, not all linguistic theories can be implemented effectively, and the knowledge engineers who work with modern ontologies usually have little or no linguistic background.

To address these issues surrounding lexical layers [1] propose a “linguistic light” approach to lexical layers for ontologies called LEON. The LEON approach proposes that the lexical layer for an ontology consists of a tuple of the form

$$\langle CitationForm, Constraints\rangle$$

for each semantic entity with a lexical realisation encoded in the ontology. The first element of the tuple, the citation form, is the basic form of the lexical realisation. The second element of the tuple is a set of constraints which specifies if and how the citation form can vary. This facilitates linguistic phenomena such as inflection and derivation as well as allowing the modelling of multi-word units which vary in both their surface form and word order using this simple approach. This approach does not focus on the linguistic description of vocabulary associated with a concept but on the linguistic features of a given concept in order to identify class instances in text. This allows for when one concept might have

[4] http://www.tc37sc4.org

[5] http://www.ilc.cnr.it/EAGLES96/browse.html

[6] http://www.mpi.nl/ISLE/

[7] General Architecture for Text Engineering

several different lexical realisations with different linguistic descriptions for example:

- New York, Big Apple, NY
- Rosetta Stone, Stone of Rosetta
- International Business Machines, IBM, Big Blue

In addition, it is deliberately less complex in order to cater for users, in particular knowledge engineers, who lack a linguistic background, but may wish to develop an ontology with linguistic features included. We note that the interlingua here is an the ontology. Therefore the design and conceptualisation used in the ontology could be a limiting factor where there are semantic divergences between languages or domain terminology.

## IV. Linguistic Light Multilingualism

The "linguistic light" paradigm for lexical layers is flexible and can be applied multi-lingually with little effort. This is because the extensions are not tied to any particular language or formalism.

### A. Extending LEON's lexical layer

In order to expand the LEON lexical layer description to cope with another language we must provide an appropriate citation form and set of constraints for the lexical realisation(s) of that concept in the new language. This second tuple can then be merged with the existing data giving rise to a tuple consisting of a set of citation forms and a set of constraint sets corresponding to each citation form.

$$\left\langle \left\{ \begin{array}{c} CitationForm_{EN} \\ CitationForm_{FR} \\ CitationForm_{DE} \\ CitationForm_{...} \end{array} \right\}, \left\{ \begin{array}{c} \{cnstr1_{EN}, cnstr2_{EN}, ...\} \\ \{cnstr1_{FR}, cnstr2_{FR}, ...\} \\ \{cnstr1_{DE}, cnstr2_{DE}, ...\} \\ \{cnstr1_{...}\ , cnstr2_{...}\ , ...\} \end{array} \right\} \right\rangle$$

The multilingual lexical layer can then be used to easily retartget the ontology to a given language or locale by using the appropriate citation forms and constraint sets.

### B. Signature Detection

Effective use of the linguistic light LEON lexical layer in text analytics and ontology-based information extraction applications relies on unstructured text being processed and the "signature" of a term mention being detected. The lexical analyser used to process the text needs to have some means of normalising variant forms to a common stem or lemma in order to be able to put forward potential signature tokens.

To ensure high recall, normalisation of constituents is important, especially for languages with more a complex morphology than English. In this paper we pay more attention to the normaliser as a component of a linguistic light solution. Given proper normalisation, we believe that the LLA/LLS approach will provide very high recall in a multilingual environment.

## V. "Linguistic Light" Normalisation

### A. Character Normalisation

This type of normalization accounts for typographic variances like using capitalisation and diacritics in Latin and Cyrillic based scripts ("*Böblingen*" vs. "*Boeblingen*"), the use of different scripts in Japanese texts, auxiliary usage of vowels in Arabic or Hebrew; regular spelling variations (British "*colour*" vs. American "*color*"). Some types of character normalisation might be efficiently performed by algorithmic methods.

### B. Morphological Normalisation

Morphology is the subfield of linguistics that studies the internal structure of words. In linguistics, two types of morphological normalization are traditionally referred to, namely lemmatization and stemming. Lemmatization accounts for inflectional variants of the same word where part of speech is preserved. For example different cases, genders, numbers (like singular form of noun *database* and its plural form *databases*).

Stemming frequently involves a more "aggressive" normalization, which accounts for both inflectional and derivational morphology, where related words are mapped onto the same index, even if they have different parts of speech. For example, one can map the words *computerization*, *computerize*, *computer*, *computing*, *compute* onto the same index. An index term can be a non-word like *comput* (a minimal and hopefully unambiguous denotation of all related terms). Stemming therefore has the effect of "conflating" the index more aggressively than lemmatization, by mapping a wider set of word forms to a single index term, thereby resulting in higher recall i.e. in any query term finding more documents during search.

### C. Synonym Normalisation

At least for some domains, if not for language in general, it might be reasonable to consider some words as exact synonyms and map them into the same index (for example, liver/hepatic, renal/kidney). Dictionaries of linguistic synonyms are not frequently used in indexing because linguistic synonyms are typically not exact synonyms (for example, using the chain of synonyms in MS-Word: average ≈ mean ≈ nasty ≈ shameful one can wrongly equate average with shameful). The quality of IR and IE (depending on the task) is characterized by two intrinsic metrics: recall (the ratio of the number of relevant documents returned to the total number of relevant documents in the collection of documents indexed) and precision (the ratio of the number of relevant documents retrieved to the total number of documents retrieved). Search engines typically trade off precision for recall. In the absence of accurate relevancy ranking algorithms the user is left to sort through extensive lists of documents for the correct information. So the challenge is to achieve high precision without significantly reducing recall.

Word normalization is essential for the quality of IR systems. Research to date indicates that some character normalization is indispensable to improve recall. Morphological normalization in general improves recall, but may degrade precision. Although stemmers are widely used by the majority of IR systems, their role for IR is frequently disputed; however, it is generally accepted that morphological normalization is indispensable for highly inflected languages (like Finnish) when the same word might have dozens of forms. It is also needed for languages with frugal morphology (like English) in the scenarios where most of the analysed documents are short. For example, if the document about databases is rather long, one might expect that the term database will be encountered in both grammatical forms: data-base and databases, and one can afford not to map both forms into the same index because the document will be retrieved as relevant to the query containing the search item database anyway. However, if the document is short, it might happen that the document will contain only mentions of plural form, in which case the document will be missed. For some time Google did not use stemming in order "to provide the most accurate results to its users". However, Google subsequently introduced stemming technology into its system "Thus, when appropriate, it will search not only for your search terms, but also for words that are similar to some or all of those terms. If you search for "pet lemur dietary needs", Google will also search for "pet lemur diet needs", and other related variations of your terms[8]."

### D. Reversed Finite State Normalisation

Following [11], which is based on the work of [12], a normaliser can be built from the lexicon by combining common suffixes in a finite state automaton. A finite state automaton (FSA) is a computational model made up of states and transitions. Given an input sequence (e.g. a word as a sequence of letters), the FSA moves through a series of states according to transitions that given a current state match the current input symbol (letter). In Figure 1 there are a number of possible input sequences that reach the final state e.g. smart, start etc. The final state can be associated with information about the sequence that leads to it, such as an algorithm that produces a normal form.

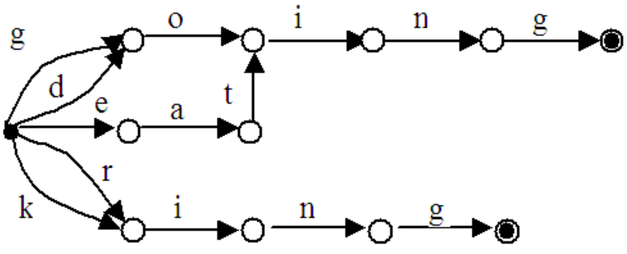


Fig. 1. Finite State Automaton

A reversed finite state normaliser is a finite state automaton which traverses the input string in reverse character order. A reversed FSA can be compiled from a full form word list, electronic dictionary or similar resource for the language or domain concerned. The resulting FSA will be such that morphological suffixes are conflated into common paths of transitions leaving word stems following branching states.

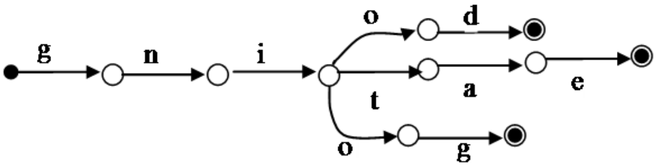


Fig. 2. Reversed Finite State Automaton

Notice that by exploiting common endings in this way the size and complexity of the FSA is reduced. This computational approach to building a normaliser does not necessarily produce proper root form lemmas for the input, instead a reduced stem is produced. These stems can often be non-word tokens but they will correspond to the orthographical root of one or more full form of the actual word it represents. These stems can then be used by a "Linguistic Light Scanner" (described in [1]) to increase recall in the identification of term mentions in text. The LEON constraints for a given citation form then determine which (if any) variants are permissible for a valid recognition.

By combining LEON and reverse FSA normalisation these two linguistically light, but computationally efficient models for lexical analysis no precision is lost because the stemming process reduces full forms to concise stems while the LEON constraints then allow or disallow inflected (or otherwise orthographically different) forms. This makes the process of adding a new language to a lexical layer relatively simple and quick to implement without any significant linguistic knowledge about the language, all that is necessary is a word list. For these reasons we suggest this type of approach to normalisation and signature detection. This approach can also deal with character normalisation where adding all variations into a full form lexicon would become unwieldy

## VI. Multilingual Issues

When dealing with identifying term mentions in multiple languages the compatibility of the lexical description with features of the various languages is an important consideration. In the previous work ([1]) the only language considered is English, which is relatively frugal with respect to morphology, casing, and agreement when compared with other languages. Other languages also have different constraints on sentential word order which can be important to detection. We will look at some examples of how these aspects of language can be problematic and how they are handled in the linguistic light paradigm.

*1) Agreement:* Many languages require that, for example, adjectives and nouns agree with respect to number, gender, case etc. So, for instance, a singular noun can only have a singular adjective used to describe it. These constraints are important regarding the grammaticality and correctness of the language. This type of constraint is not enforced in the LEON

[8] Taken from http://www.google.com/help/basics.html

model. However, as the following example shows, it is often beneficial not to enforce such linguistic constraintsas to do so would affect recall where there has been a human error, or a deliberate mistake owing to creative licence.

Take the French term “Intelligence Artificielle,” in this example, the gender and number agreement of the two tokens is obligatory. If it occurs with a disagreement, then it is likely a human typing mistake like “intelligence artificiel.” In this case, the disagreement is a typing error, as there is a disagreement between the noun (“intelligence”: singular feminine) and the adjective (“artificiel”: singular masculine). This can occur in texts, and it will be detected if the exact string match is turned off (to allow infleced variants of the citation form). However it would be missed if the agreement constraint were to be strictly enforced. This also allows the detection of instances in other contexts. For example, “vie et intelligence artificielles”, where “artificielles” disagrees in gender and number because here it refers to two entities which are “vie” and “intelligence.”

Likewise in Russian gender agreement is a present and important feature for grammatical correctness, however if we take the term “sistemnuj administrator” (system administrator) where both terms are in the masculine, and change one to feminine like

systemnaja administrator
Adj Masc Noun Masc

This ungrammatical noun phrase yields a single hit in a search on Google’s index.[9] The text in which the example was found is using the gender disagreement as a subtle device to highlight that the person in question is woman and draw attention to this fact.

*2) Word Order:* Some languages have a less rigid restriction on sentential word order than others, German for example has quite strict rules regarding word order, Russian on the other hand is less so. This needs to be considered with regards detecting MWU lexical realisations in text analysis. A language with a freer word order means there are more possible ways of constructing a sentence which refers to a given concept. Therefore, in theory, the search space is larger and correspondingly so is the likelihood of detecting false positives.

Consider the French MWU “Maladies Sexuellement Transmissibles” (sexually transmitted diseases). If we encounter the same words in varying order and forms like “maladie mortelle sexuellement transmissible,” and “Maladie transmise sexuellement” the underlying concept which is being referred to remains the same. So a language with a freer word order is not necessarily a problem for MWU lexical realisations.

## VII. CONCLUSIONS

We have examined a number of linguistic considerations for ontology lexicalisation across multiple languages. We have also discussed the LEON “linguistic light” approach to adding a lexical layer and shown how it is robust enough to handle various linguistic nuances without having to explicitly encode linguistic information. The caveat, however, is that in order to detect the linguist “signatures” of term mentions in text the LEON approach needs some suitable normalisation of the input text.

Following in the linguistic light vein we have shown how a simple, robust normaliser can be induced from a wordlist in the form of a reverse finite state automaton. Once the lexical layer for an ontology has been implemented, the appropriate wordlist already exists in the form of the citation form lexical realisations encoded in the lexicon, so a reverse FSA normaliser can be rapidly produced for the appropriate vocabulary. By combining these two linguistic light approaches to analysing natural language in text an ontology can be rapidly retargeted to a new language or domain with little or no linguistic information or expertise other than an appropriate vocabulary.

## ACKNOWLEDGMENTS

The work presented in this paper was supported (in part) by the European project NEPOMUK No. FP6-027705 and (in part) by the Lion project supported by Science Foundation Ireland under Grant No. SFI/02/CE1/I131.

## REFERENCES

[1] B. Davis, S. Handschuh, A. Troussov, J. Judge, and M. Sogrin, “Linguistically Light Lexical Extensions for Ontologies,” in *Proceedings of LREC 2008*, Marrakech, Morrocco, 2008.

[2] M. Espinoza, A. Gómez-Pérez, and E. Mena, “Enriching an ontology with multilingual information,” in *ESWC*, 2008, pp. 333–347.

[3] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano, “LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies,” in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, 2006.

[4] G. Francopoulo, M. George, N. Calzolari, M. Monachini, N. Bel, M. Pet, and C. Soria, “Lexical Markup Framework (LMF),” in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, 2006.

[5] A. Gomez-Perez, O. Corcho, and M. Fernandez-Lopez, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition (Advanced Information and Knowledge Processing)*. Springer, July 2004. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/1852335513

[6] G. A. Miller, “Wordnet: a lexical database for english,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[7] Piek Vossen, Ed., *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[8] Wim Peters and Piek Vossen and Pedro Díez-Orzas and Geert Adriaens, “Cross-linguistic Alignment of Wordnets with an Inter-Lingual-Index,” pp. 149–179, 1998.

[9] R. Cilibrasi and P. M. B. Vitanyi, “The google similarity distance,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, p. 370, 2007. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0412098

[10] D. Maynard and H. Cunningham, “Multilingual adaptations of annie, a reusable information extraction tool,” in *EACL ’03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 219–222.

[11] A. Troussov and B. O. Donovan, “Morphosyntactic Annotation and Lemmatization Based on the Finite-State Dictionary of Wordformation Elements,” in *Proceedings of Speech and Computer (SPECOM)*, Moscow, Russia, 2003, pp. 27–29.

[12] J. Daciuk, “Incremental Construction of Finite-state Automata and Transducers, and their Use in the Natural Language Processing,” Ph.D. dissertation, Technical University of Gdansk, 1998.

[9] Search performed on June 25th 2008

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Modeling the Frequency of Phrasal Verbs with Search Engines

Grażyna Chamielec
SuperMemo
Poznan, Poland
ika.chamielec@gmail.com

Dawid Weiss
Institute of Computer Science
Poznan University of Technology
Poznan, Poland
dawid.weiss@cs.put.poznan.pl

***Abstract*—There are well over a thousand phrasal verbs in English. For non-native speakers they are notoriously difficult to remember and use in the right context. We tried to construct a ranking of phrasal verbs according to their estimated occurrence frequency, based on quantitative information available from the public indexable Web. Technically, we used major Web search engines to acquire phrase-occurrence statistics, measured consistency between the rankings implied by their results and confirmed that a rough set of 'classes' of phrasal verbs can be distinguished.**

**While this technique relies on inaccurate and possibly biased estimation functions, we show that the overall distribution of ranks seems to be consistent among all the queried search engines operated by different vendors.**

## I. Introduction

A *PHRASAL verb* is, according to Oxford Advanced Learner's Dictionary [1]:

> [. . . ] a simple verb combined with an adverb or a preposition, or sometimes both, to make a new verb with a meaning that is different from that of the simple verb, e.g., *go in for*, *win over*, *blow up*.

There are a number of phrasal verbs in both spoken and written English ([2] lists over 6000 entries). As the definition states, the meaning of a phrasal verb cannot be easily guessed from individual components—many non-native speakers of English must therefore memorize phrasal verbs in order to be able to understand and use them in the right context. Our motivation for this work was a direct consequence of this observation.

SuperMemo[1] is a company specializing in helping people learn fast, use memory efficiently and aid in self-improvement processes. SuperMemo's line of products include, among others, dictionaries and language courses. While working on a list of English phrasal verbs, we stated the following problem:

- Which phrasal verbs should be memorized first?

There are two other related questions:

- Are there any phrasal verbs that are hardly ever present in a 'live' corpora of written language?
- Are there groups of 'frequent' and 'infrequent' phrasal verbs and is it possible to distinguish these groups?

There is certainly no definite answer to these questions; phrasal verbs and their meaning will vary by region and dialect of English, for example. Our research intuition was telling us though, that by relying on a really large corpora of existing texts rather than book resources or dictionaries, we could come out with a fairly good estimate on which phrasal verbs are common and which are infrequent. In other words, we wanted to measure possibly 'real' average occurrence frequency of each phrasal verb, then sort them in the order of this estimated frequency and distinguish several groups that could provide the basis for the construction of a training course.

## II. Related work and discussion

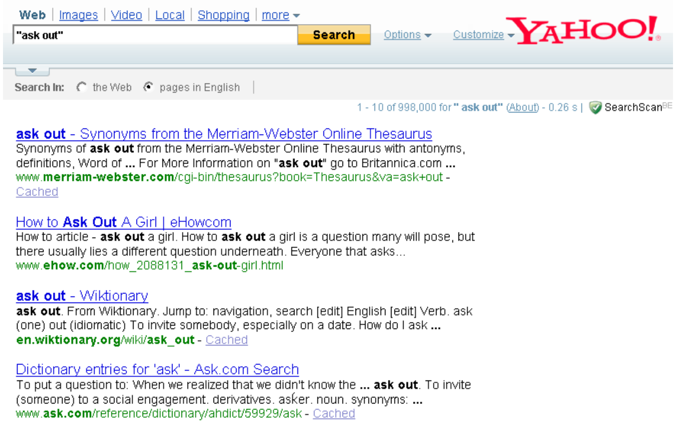
There exist a number of dictionaries [2], [3], books and papers concerning phrasal verbs and verb-particle associations at the linguistic layer. There are also on-line resources listing phrasal verbs and providing their meanings. However, we failed to find any resource that would attempt to *quantitatively* measure the frequency of use of phrasal verbs. The paper by Timothy Baldwin and Aline Villavicencio came closest to our expectations [4]. In this work, authors process raw text of the Wall Street Journal corpus using a number of different methods to identify verb-particle occurrences. The best technique reached the f-score of 0.865. The experiment in [4] was performed on an established corpus of press resources. While using a corpus like this (or a balanced language corpus in general) has many advantages, we wanted to stick to the Web because it reflects many different language users, use cases and is a great deal larger than any other corpus available. Although there are various opinions about the coverage of the Web, its information quality and bias (see [5] or [6] for an interesting discussion), we believe that in our case these aspects can be neglected and search engines provide suitable source of knowledge to answer the questions given in the introduction. Obviously, any research based on uncontrolled, proprietary information sources such as search engines should be approached with care. We tried to do our best to cross-validate the results against multiple vendors to make them more confident.

## III. Proposed methodology

Every search engine returns an estimation of the number of documents 'matching' a given query (note that this is the number of *documents*, not individual instances of the query). Figure 1 illustrates a query results page with the rough number

[1] http://www.supermemo.com

978-83-60810-14-9/08/$25.00 © 2008 IEEE


Web | Images | Video | Local | Shopping | more
"ask out"
Search
Options
Customize
YAHOO!
Search In: ○ the Web ● pages in English
SearchScan

1 - 10 of 998,000 for " **ask out**" (About) - 0.26 s |

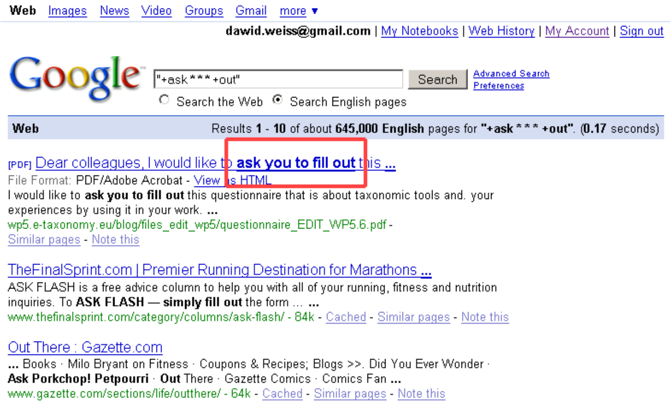
**ask out** - Synonyms from the Merriam-Webster Online Thesaurus
Synonyms of **ask out** from the Merriam-Webster Online Thesaurus with antonyms, definitions, Word of **...** For More Information on "**ask out**" go to Britannica.com **...**
www.**merriam-webster.com**/cgi-bin/thesaurus?book=Thesaurus&va=ask+out - Cached

How to **Ask Out** A Girl | eHowcom
How to article - **ask out** a girl. How to **ask out** a girl is a question many will pose, but there usually lies a different question underneath. Everyone that asks...
www.**ehow.com**/how_2088131_**ask-out**-girl.html

**ask out** - Wiktionary
**ask out**. From Wiktionary. Jump to: navigation, search [edit] English [edit] Verb. ask (one) out (idiomatic) To invite somebody, especially on a date. How do I ask **...**
**en.wiktionary.org**/wiki/**ask_out** - Cached

Dictionary entries for 'ask' - Ask.com Search
To put a question to: When we realized that we didn't know the **...** **ask out**. To invite (someone) to a social engagement. derivatives. asker. noun. synonyms: **...**
www.**ask.com**/reference/dictionary/ahdict/59929/ask - Cached

Fig. 1. Query results page from Yahoo search engine. The red rectangle marks the status line displaying the number documents matching the query.


Web Images News Video Groups Gmail more ▾
My Notebooks | Web History | My Account | Sign out
Google
"+ask * * * +out"
Search
Advanced Search
Preferences
○ Search the Web ● Search English pages
Web

**dawid.weiss@gmail.com**

Results **1** - **10** of about **645,000 English** pages for **"+ask * * * +out"**. (**0.17** seconds)

[PDF] Dear colleagues, I would like to **ask you to fill out** this **...**
File Format: PDF/Adobe Acrobat - View as HTML
I would like to **ask you to fill out** this questionnaire that is about taxonomic tools and. your experiences by using it in your work. **...**
wp5.e-taxonomy.eu/blog/files_edit_wp5/questionnaire_EDIT_WP5.6.pdf - Similar pages - Note this

TheFinalSprint.com | Premier Running Destination for Marathons **...**
ASK FLASH is a free advice column to help you with all of your running, fitness and nutrition inquiries. To **ASK FLASH — simply fill out** the form ... **...**
www.thefinalsprint.com/category/columns/ask-flash/ - 84k - Cached - Similar pages - Note this

Out There : Gazette.com
**...** Books · Milo Bryant on Fitness · Coupons & Recipes; Blogs >>. Did You Ever Wonder · **Ask Porkchop! Petpourri** · **Out** There · Gazette Comics · Comics Fan **...**
www.gazette.com/sections/life/outthere/ - 64k - Cached - Similar pages - Note this

Fig. 2. A wildcard query may result in a false match (see the marked phrase).

of documents matching the exact phrase ‘ask out’. While the returned number is merely an estimate and may be inaccurate (we discuss this in Section V), we assumed that the estimation is correct at least to the order of magnitude, thus properly dividing frequent and relatively infrequent phrasal verbs.

## IV. Phrasal Verbs Considered

We used a hand-crafted set of phrasal verbs (PV) collected from several on-line resources and books. We made an explicit distinction between separable and inseparable PVs where it was appropriate and placed an asterisk (wildcard) character in places where separation could occur. Then, to every verb we assigned a number of different forms in which it could possibly appear in the text, depending on its tense. Table 3 illustrates an example phrasal verb pattern and all its corresponding variations. The pattern-based representation was used to drive queries to search engines.

## V. Potential Ambiguities and Other Problems

There are a few corner cases in counting the number of documents containing a given phrasal verb and they are all a consequence of how text information retrieval methods (implemented in search engines) work. In simplest terms, search engines transform a document into a vector of individual words and their *weights* (relative importance of a given word to the document). This representation of text is called the vector space model [7]. A query to a search engine returns all documents that contain a union of the query’s set of words (possibly ordered), but it is rarely possible to specify deeper contextual constraints. Let us explain the possible side-effects of this process on a few examples.

The first problem is that not every word pattern corresponds to an actual phrasal verb. For example, *[to] be in* can appear as *I’m in*, but the sole appearance of this sequence of words without the knowledge of the context may be a false hit (*I’m in Poland right now.*). Unfortunately this will be the case with most verbs that have transitive and intransitive forms. Another issue is caused by multiple meanings of a single phrasal verb, compare *throw up* (vomit) and *throw up* (an idea). Detecting and separating the meaning of these two expressions seems impossible assuming the measurement technique we agreed to use.

The final example concerns separable forms of phrasal verbs. What we intend to do is to query for patterns (sequences of words) that have a few words in between (but not too many). For example, *sign me in* should be counted as an occurrence of *sign in*. However, simply allowing words to appear in between components of a phrasal verb may lead to many mistakes. For instance, as Figure 2 illustrates, the three top-ranked documents for a query *ask out* separated by three other words, are basically wrong. There seems to be no way of filtering out this noise without more complex linguistic analysis (if we had access to whole document content, as in a controlled corpus, we could use POS tags for getting rid of such errors).

Regardless of the above problems, we decided to calculate occurrence statistics and proceed with the experiment. It is our assumption that the number of false matches for less than three wildcards can be neglected compared to the number of true matches (at least for common phrasal verbs). As for phrasal verbs with multiple meanings, all occurrences of these meanings sum up to one figure which reflects the aggregated use of a given sequence of words. Since so, the final ranking

| | | | |
|---|---|---|---|
| **ask/asks/asked/asking * out** | | | |
| ask out | ask – out | ask – – out | ask – – – out |
| asks out | asks – out | asks – – out | asks – – – out |
| asked out | asked – out | asked – – out | asked – – – out |
| asking out | asking – out | asking – – out | asking – – – out |
| **back/backs/backed/backing off** | | | |
| back off | backs off | backed off | backing off |
| **crack/cracks/cracking/cracked * up** | | | |
| crack up | crack – up | crack – – up | crack – – – up |
| cracks up | cracks – up | cracks – – up | cracks – – – up |
| cracked up | cracked – up | cracked – – up | cracked – – – up |
| cracking up | cracking – up | cracking – – up | cracking – – – up |

Fig. 3. An example of phrasal verb patterns and matching word sequences. An asterisk (*) symbol represents between zero and three words appearing in its position, we denoted these words using the dash symbol on the right (–).

position is to some extent indicative of the need to learn a given phrasal verb (even if it is ambiguous).

## VI. Collecting occurrence statistics

We collected occurrence statistics from several search engines: Google (`www.google.com`), Yahoo (`www.yahoo.com`), AllTheWeb (`www.alltheweb.com`), Gigablast (`www.gigablast.com`) and Microsoft Live (`www.live.com`). With the exception of Gigablast and Microsoft Live, the remaining providers all support the so-called *wildcard queries*, i.e., a query for all documents containing a given phrase separated by one or more unrestricted words inside. With wildcard queries we could estimate the number of occurrences of separable phrasal verbs by querying for the exact phrase, phrase with one, two and three extra words at the point of possible separation. For example, the entry (to simplify, we only show one verb form here):

```
ask * out
```

would result in the following queries to a search engine:

```
ask out
ask * out
ask * * out
ask * * * out
```

An exact format of queries submitted to each search engine varied depending on the service provider's syntax and we omit it here, although we found out that such details are quite crucial because search engines employ various optimizations and query expansion techniques that, in our case, distorted the output. As previously observed in [5], the returned estimation counts have some significant variance within the same query (the same search engine would return a different document count for consecutive executions of an identical query). We took this into account and put together 10 identical query lists, randomized their order and executed all queries at different times and from different machines. Finally, we paid particular attention to restricting the search to documents in the English language and to searching within document content only (exclude links pointing to the page).

The process of querying search engines was partially automated and performed in accordance with each search engine's policies and terms of use specifications (timeouts between queries, use of automated programming interfaces when possible).

## VII. Results

Overall, we collected frequency counts for 10 633 various separable and inseparable forms of 991 phrasal verb patterns (some of these were closely related, like *blend in* and *blend into*). For each form, we stored the estimated document count for each of the 10 'samples' made to each single search engine. Even though the querying process was semi-automatic, it lasted over three days (because we had to add the required timeouts between queries) and involved over 30 machines (with different IP addresses). If we had been given access to the search engine's infrastructure, such processing could be made much faster and more accurately by running shallow grammar parsing on the content of each document, splitting the process over multiple machines using the map-reduce paradigm.

We describe the results from several angles in sub-sections below.

### A. Differences between search engines

Every search engine is a bit different—these differences usually concern the number of indexed documents, ranking algorithms and technical aspects of estimating the number of matching documents. Our first step was to cross-compare the numbers returned from various search engines to see if they share similar distribution and what shape this distribution is.

We took one sample out of the ten made and for each phrasal verb form we compared document counts between search engines by sorting all forms according to the number of documents returned by Yahoo, placing them (in this order) on the horizontal axis and plotting document counts on the vertical axis. Figures 4–7 demonstrate the results. The overall distribution shape for all search engines is for the most part exponential (vertical axis is on logarithmic scale). Exponential distribution confirms our initial intuition that a small number of phrasal verbs occurs frequently and a great deal of them are relatively infrequent on the Web.

Back to differences between search engines, we can observe notable differences in average document counts between different search engines, but highly correlated distribution shapes. This validates our assumption that search engines are a methodologically sound tool to 'probe' the Web. If (theoretically) we consider the Web to be a global population of documents, then the index of each search engine is basically a random sample taken from this population. If so, the average count of documents between two search engines should be linearly proportional to the degree of a constant multiplier. Another way to put it is that the *ordering* of phrasal verb forms imposed by all search engines should be very similar between search engines. A look at Figures 4–7 and especially at log-log plots in Figure 8 reveals that all search engines returned correlated results. For example, Yahoo and AllTheWeb's results are almost identical (Figure 4) because AllTheWeb's index is powered by Yahoo; minor differences may be a result of different search query routing inside Yahoo's infrastructure. There is also an evident high similarity between Yahoo, Gigablast and Microsoft Live's results (see log-log plots in Figure 8, although Microsoft and Gigablast have an order of magnitude smaller index. The only visibly different engine is Google—not only has it fewer documents compared to Yahoo, but also its count distribution is strikingly different compared to other search engines (although still correlated). Narrowed to only non-wildcard forms, the distribution difference is even more strange because it shows two different 'traces' of frequency distribution in the area of more frequent phrasal verbs (see Figure 8).

We initially thought this difference in Google's case might be caused by the fact that it has the largest infrastructure and queries may be routed to separate index sections, leading to

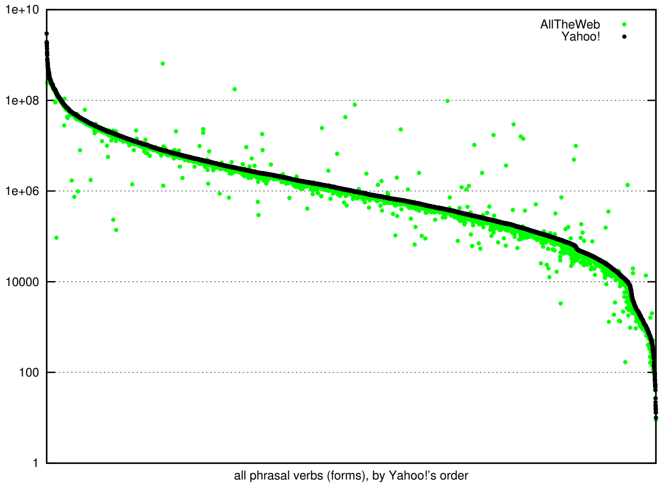


Fig. 4. Document counts for results acquired from AllTheWeb and Yahoo (sorted by Yahoo's results—the black line).

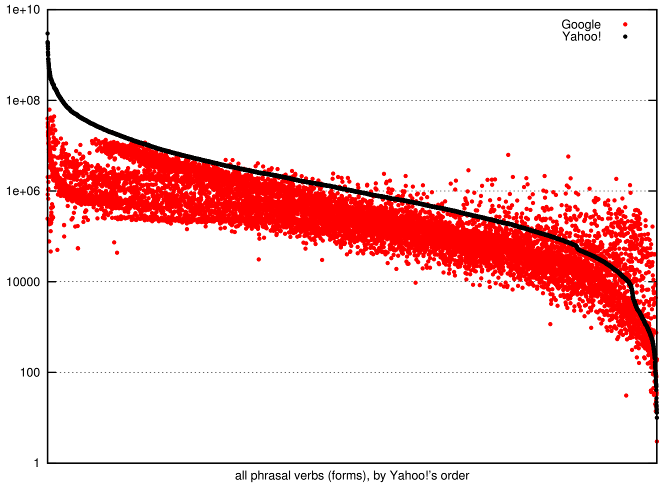


Fig. 5. Document counts for results acquired from Google and Yahoo (sorted by Yahoo's results—the black line).

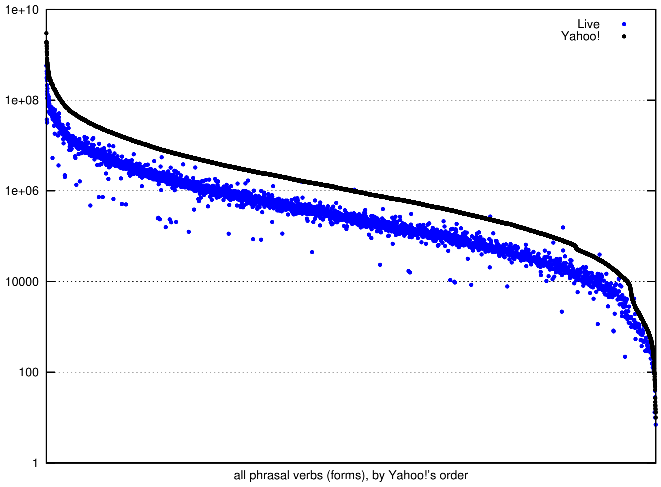


Fig. 6. Document counts for results acquired from Microsoft Live and Yahoo (sorted by Yahoo's results—the black line).

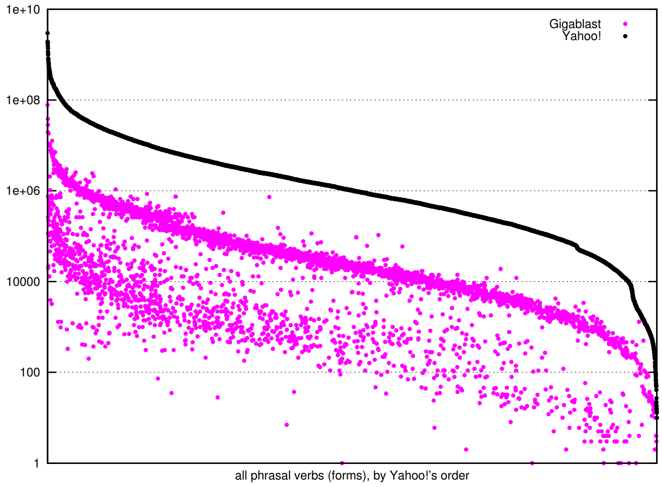


Fig. 7. Document counts for results acquired from Gigablast and Yahoo (sorted by Yahoo's results—the black line).

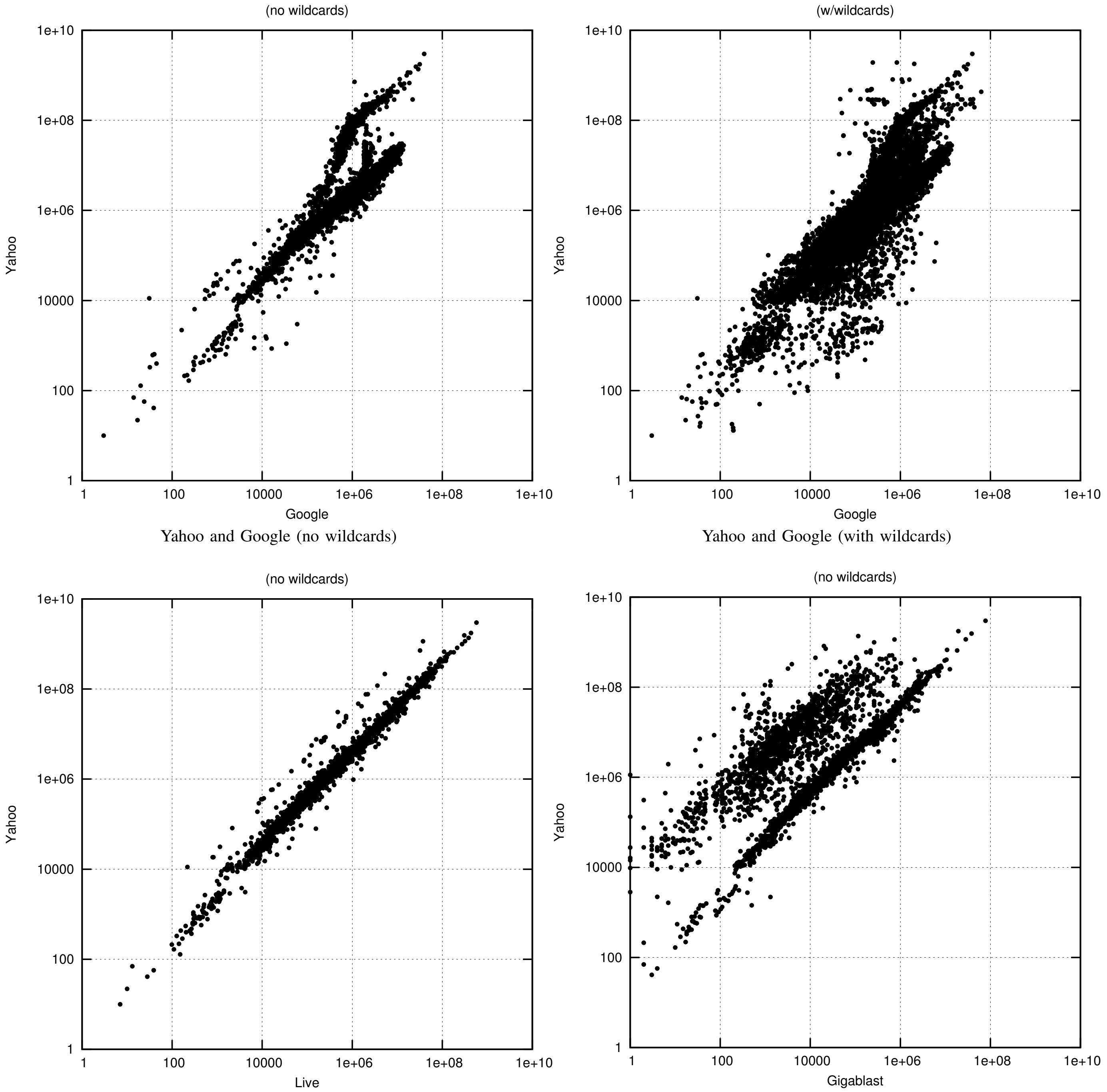
different estimated count of results. We took a closer look at all ten samples for each query, calculating minimum, maximum, median and a truncated average (average of 6 samples after sorting and removing two minimum and maximum outliers). The outcome of this analysis is that, again, Google has the largest variation between estimated result count for a single query (refer to technical report [8] for a more in-depth analysis). In case of Yahoo the difference between minimum and maximum number of results is relatively small, usually the same. Microsoft Live returns a fairly consistent range of difference—usually in the order of magnitude—with the truncated average usually equal to the maximum. For Google, the difference between min and max is again the order of magnitude, but the average is less predictable and is usually in between min and max (see Figure 9).

### B. *Phrasal verb rankings (groups)*

We constructed a *ranking* of phrasal verbs according to their totaled frequency of occurrence on the Web. Note that actual positions in this ranking are a product of multiple heuristics and their values should not be compared directly. The overall ordering should merely help to distinguish subgroups of frequent and infrequent phrasal verbs, as was our initial motivation for this research.

We experimented with many different ways of aggregating information from all samples and forms of each phrasal verb. We produced multiple possible rankings based on the following algorithm steps:

1) for every search engine, aggregate all samples for each phrasal verb form `form_id`, calculate minimum, maximum, median and truncated average (`avg2`) from document counts;
2) consider all variations: forms with <= 0, 1, 2 and 3 wildcards;



Yahoo and Google (no wildcards)

Yahoo and Google (with wildcards)

Yahoo and Live (with wildcards)

Yahoo and Gigablast (no wildcards)

Fig. 8. Log-log plot of document counts between Yahoo and other search engines.

3) sort in descending order all forms according to minimum, maximum, median and avg2 column, assign a rank to each `form_id`;
4) assign a minimum rank of any of its forms to each phrasal verb `pv_id`.

The above procedure has several variables which cause numerous possible variations of output rankings (depending on the engine, number of wildcards and the order column being considered). These rankings, consistently with our previous observations, demonstrate close similarity to each other within a single search engine and between Yahoo, AllTheWeb and Microsoft Live. Only Google is an exception. To give a few examples, the choice of the sorting column did not have much impact on the actual ranking within a single search engine. Cross-engine ranking consistency is shown on plots in Figure 10. The correlation of ranks (measured with correlation coefficient, which in this case equals to Spearman's rank coefficient) between AllTheWeb, Yahoo and Microsoft Live

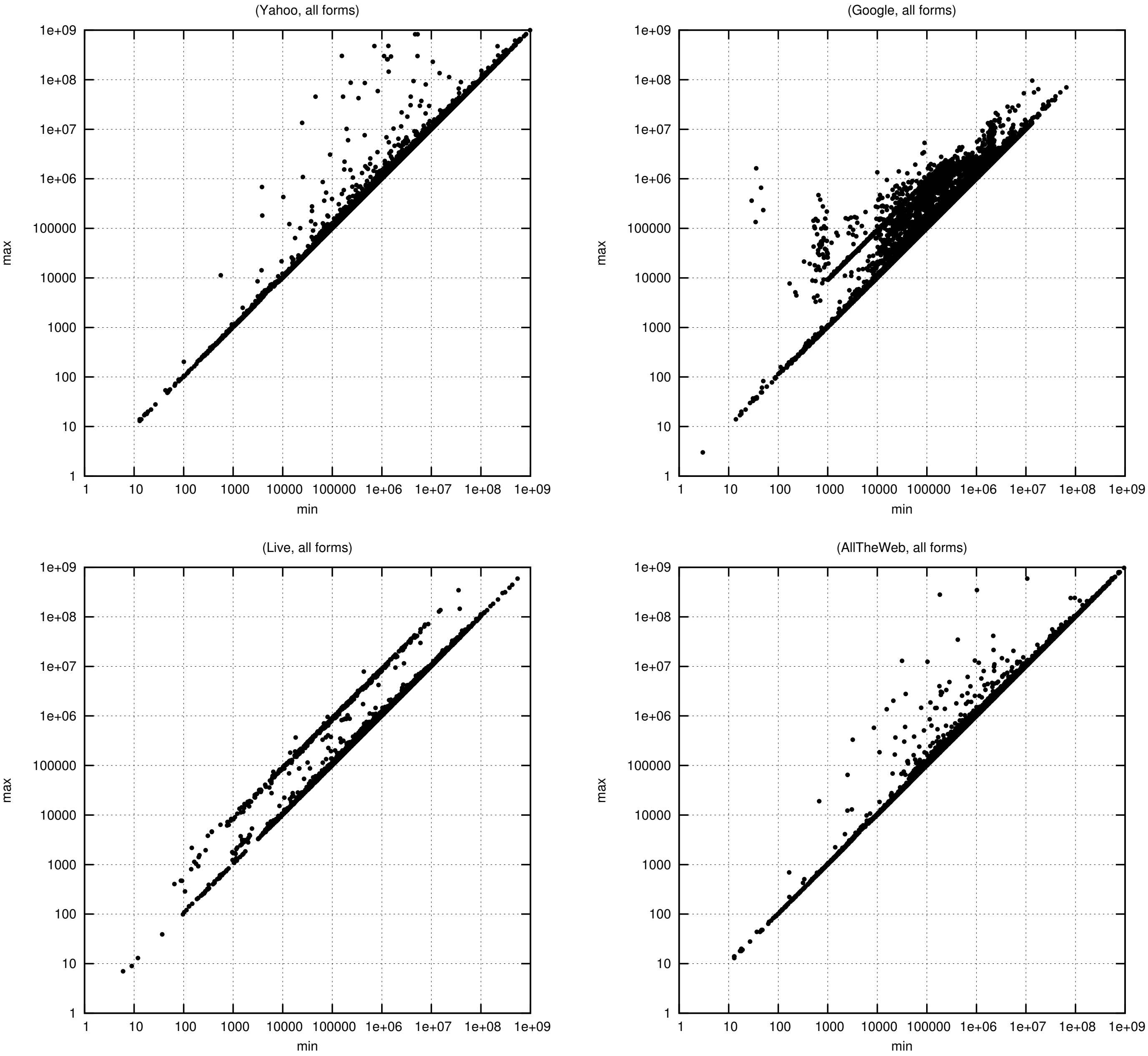


Fig. 9. Relationship between minimum and maximum number of results out of 10 samples for each phrasal verb form (Yahoo, Google, Microsoft Live and AllTheWeb).

was evident and larger than 0.9 for all considered combinations of rank computations. Google is distinctly different from other search engines, but the correlation coefficient is still quite high—between 0.7 and 0.8. We have no clear explanation as to why Google's results turn out to be slightly different than obtained from other search engines.

Even though all rankings were highly correlated, they were still a bit different from each other, so there is no ultimate one answer to our initial question of 'frequent' and 'infrequent' phrasal verbs. Without a doubt the rankings themselves reflect the nature of Web resources (see Table I) by, e.g., boosting phrases common in e-commerce (*sign up*, *check out*). Yet, a tentative and subjective feeling is that the top entries are indeed something that every native user of English should be familiar with and bottom ranking entries are extremely rare, uncommon or denote mistakes in the data set (see Table II).

## VIII. Summary and conclusions

We tried to create a ranking of phrasal verbs according to their frequency of actual use on the Web. We designed and performed a computational experiment, measuring estimated document count using several independent search engines. We think the outcomes are interesting from two different viewpoints: the linguistic one and the one concerning (dis)similarities across contemporary search engines, which turn out to be quite intriguing.

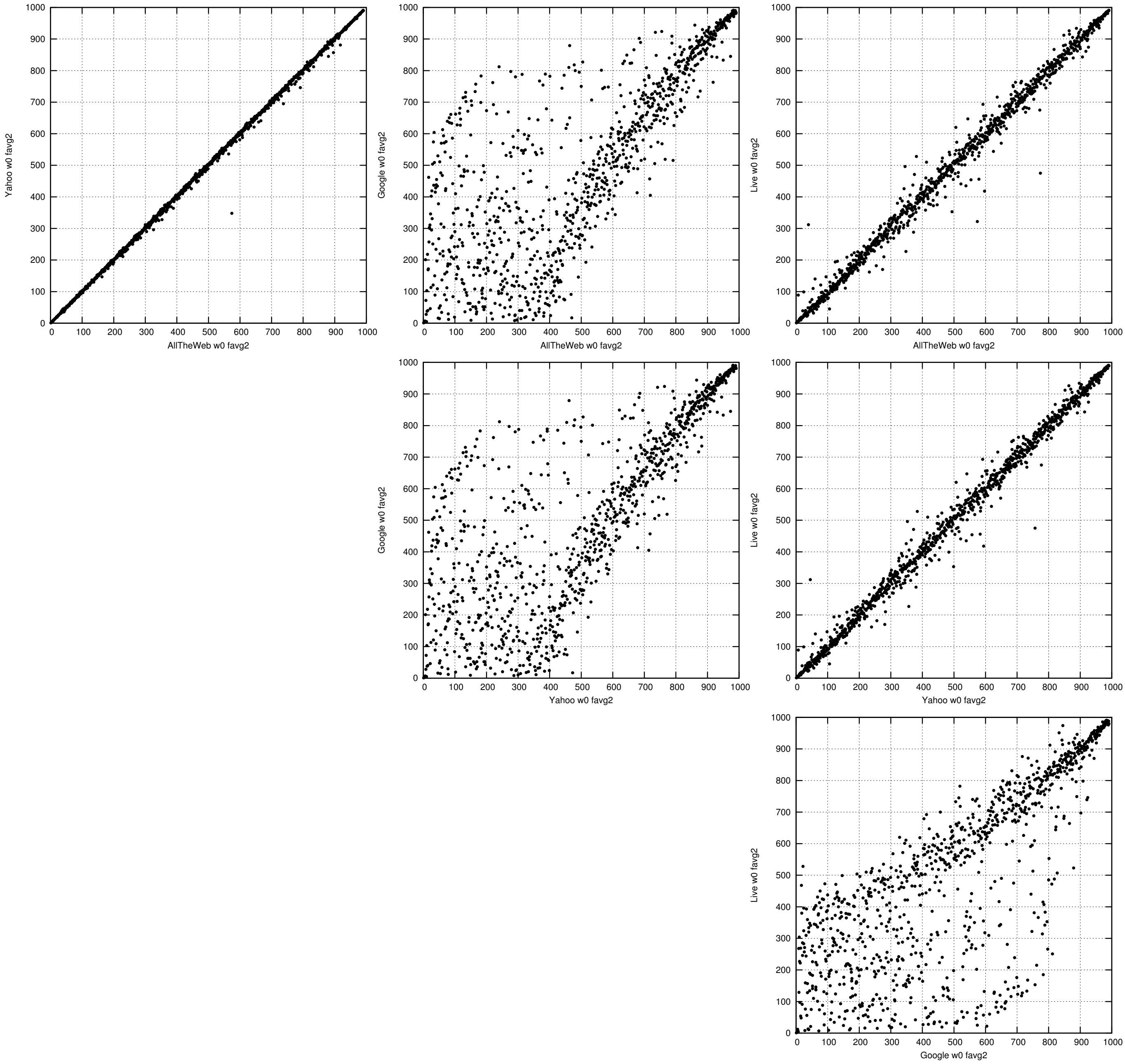


Fig. 10. Relationship between phrasal verb ranks depending on the search engine (search engine on horizontal and vertical axes fixed for rows and columns). Other parameters fixed to: `avg2` column used for sorting, zero wildcards.

As for the linguistic aspect, we are not aware of such search engine based measurement of the frequency of phrasal verbs, although search engines have been used for conducting linguistic experiments before. We think there are clear indications to believe that such an analysis can yield valid results, allowing one to separate frequent and infrequent phrasal verbs. A number of challenging problems remain unsolved:

- Even though the Web is very large, it is also biased; especially phrases that relate to e-commerce are boosted high up the ranking (*sign up*, *check out*). In our case this was not a problem because the rankings (groups) were edited manually for the final application after they were acquired anyway, but in other scenarios this is a problem.
- We currently see no way of disambiguating multi-sense phrasal verbs or no-object phrasal verbs. Given access to the full content of search engine's documents, shallow NLP techniques could be employed here.
- We used wildcard queries and multiple tenses for fetching various potential forms of phrasal verbs. It turned out that this had very little influence over final rankings; is such a step necessary or would it be enough

TABLE I
TOP 10 PHRASAL VERBS ACCORDING TO YAHOO, GOOGLE AND LIVE (0 WILDCARDS, AVG2).

| No. | Yahoo | Google | Live |
|---|---|---|---|
| 1 | sign up | sign up | sign up |
| 2 | look for | look for | look for |
| 3 | check out | be in | be in |
| 4 | be in | check out | check out |
| 5 | look at | go back | find out |
| 6 | find out | look at | look at |
| 7 | arise from | find out | set up |
| 8 | come to | be after | come to |
| 9 | set up | look in | get to |
| 10 | go back | start off | work on |

TABLE II
SELECTED 10 PHRASAL VERBS FROM THE BOTTOM OF THE RANKING FOR YAHOO, GOOGLE AND LIVE (0 WILDCARDS, AVG2).

| Yahoo | Google | Live |
|---|---|---|
| slug out | sob out | fur up |
| winkle out | slog out | suture up |
| fur up | swirl down | ravel out |
| satire up | nestle up | sponge down |
| skirt round | fur up | push round |
| sponge down | rein back | hiss off |
| tail away | skirt round | slog out |
| be bombed out | sponge down | rap put |
| slog out | ravel out | scorch along |
| hiss off | scorch along | stream down upon |

to just limit the analysis to present-tense forms?

- The distribution of document counts returned from search engines is exponential, so one could make groups of phrasal verbs each falling into bins related to the frequency's order of magnitude. However, there is no clear dividing line between these bins and there is certainly some room for improvement here.

From the point of view of a researcher interested in search engines, this work provides an interesting insight into differences between major search providers, especially with regard to the estimated matching document set size.

- Yahoo is by far the most *consistent* search engine and its returned estimation does not vary much between the same queries issued at different times,
- Yahoo and Microsoft Live show very correlated counts—nearly identical, in fact. This follows our intuition about 'sample from a large corpus', but is contradicted by results returned by Google. We cannot explain why Google is so much different compared to Yahoo and Live.
- Google and Live return document counts (for the same query) that vary by an order of magnitude.

As for further work on this subject, it would be quite interesting to examine phrasal verb distribution using exact NLP methods (or shallow, but with linguistic context taken into account) on a larger free corpora (such as Wikipedia or a free crawl of the Web) and compare the rankings with those we acquired from search engines. Such effort would allow validating and deriving further conclusions concerning the accuracy of our method. Alternatively, one could try to estimate the estimation error by taking the results returned from a search engine, manually tagging the returned documents as false/ true matches and then establishing true/false hit ratio. This method is used successfully in software engineering to establish the true number of software defects given a number of unreliable referees assessing code quality. Access to input lists of phrasal verbs, crawl results and rankings is given at the following address: http://www.cs.put.poznan.pl/dweiss/research/pv/.

**Acknowledgment** We are very grateful for anonymous reviewer feedback and his or her constructive suggestions and comments. The work was carried out with the financial support of Polish Ministry of Science and Higher Education.

## REFERENCES

[1] *Oxford Advanced Learner's Dictionary*. Oxford University Press, 1995.
[2] *Oxford Phrasal Verbs Dictionary for Learners of English*. Oxford University Press, 2007.
[3] *Cambridge Phrasal Verbs Dictionary*. Cambridge University Press, 2006.
[4] T. Baldwin and A. Villavicencio, "Extracting the unextractable: a case study on verb-particles," in *COLING-02: proceedings of the 6th conference on Natural language learning*. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 98–104.
[5] A. Kilgarriff, "Googleology is bad science," *Computational Linguistics*, vol. 33, no. 1, pp. 147–151, 2007.
[6] N. L. Waters, "Why you can't cite wikipedia in my class," *Communications of the ACM*, vol. 50, no. 9, 2007.
[7] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
[8] G. Chamielec and D. Weiss, "Modeling the frequency of phrasal verbs with search engines," Institute of Computing Science, Poznan University of Technology, Poland, Technical Report RA-05/08, 2008.

# 8th International Multidisciplinary Conference on e-Commerce and e-Government

WE WOULD like to invite original papers on all aspects of electronic commerce, electronic government and related issues. The conference is meant to address the academic community as well as representatives of business, industry, government, NGOs and information technology sector. Thus—apart from research papers—practical presentations of existing solutions are very welcome, too. This year we specifically added the EGOV component as we would like to explore the e-government research and applications.

The areas of e-Commerce, e-Governance, e-Government etc. are interdisciplinary by nature: they involve people with their background in economy, management, artificial intelligence, computer science, sociology, psychology, law and so on. We feel that a meeting of specialists in those areas may help to cross the boundaries between the traditional disciplines, but also between the communities of theorists and practitioners. And to create a better understanding of the mechanisms underlying electronic markets, electronic administrations and networked organizations.

The general list of topics includes:

- electronic markets and electronic marketing
- business models and processes in e-Commerce and e-Government
- languages and models for e-Commerce and e-Government
- artificial Intelligence in e-Commerce and e-Government
- electronic contracting and public procurement
- legal aspects of e-Commerce and e-Government
- electronic interaction and negotiation
- virtual enterprises and knowledge management
- technology for e-Commerce and e-Government
- Internet computing, networked enterprises and networked governments
- social aspects of e-Commerce and e-Government
- futurology of e-Commerce and e-Government
- e-Inclusion and its influence on e-Commerce and e-Government
- national and cross-boarder e-Government services
- productivity and efficiency in e-Commerce and e-Government

## International Programme Committee

**Cene Bavec,** University of Primorska, Slovenia

**Leszek Borzemski,** Wrocław University of Technology, Poland

**Kyril Boyanov,** Institute for Parallel Processing, Bulgarian Academy of Sciences, Bulgaria

**Wojciech Cellary,** Poznań University of Economics, Poland

**Jen-Yao Chung,** IBM T. J. Watson Research Center, USA

**Jan Maciej Czajkowski,** University of Łódź, Poland

**Krystyna Doktorowicz,** University of Silesia, Poland

**Andrzej Florczyk,** Poland

**Matjaz Gams,** Jožef Stefan Institute, Slovenia

**Krzysztof Głomb,** „Cities on Internet" Association, Poland

**Jerzy Grzymała-Busse,** University of Kansas, USA

**Ying Huang,** IBM T. J. Watson Research Center, USA

**Wolfgang Kleinwächter,** University of Aarhus, Denmark

**Václav Matyáš,** Masaryk University, Czech Republic

**Mieczysław Muraszkiewicz,** Warsaw University of Technology, Poland

**Oleksii Oletsky,** National University of Kyiv-Mohyla, Ukraine

**Olle Olsson,** Swedish Institute of Computer Science, Sweden

**David Osimo,** Institute for Prospective Technological Studies, European Commission, Spain

**Anne-Marie Oostveen,** Oxford Internet Institute, Oxford University, United Kingdom

**Arvo Ott,** e-Governance Academy, Estonia

**Andrew Pinder,** Becta, United Kingdom

**Andreja Pucihar,** University of Maribor, Slovenia

**Chunming Rong,** University of Stavanger, Norway

**Tomas Sabol,** Technical University of Košice, Slovak Republic

**Bolesław Szafrański,** Military University of Technology, Poland

## Organizing Committee

**Borys Czerniejewski,** Institute of Innovation and Information Society Ltd., Poland

**Jacek Wachowicz,** Gdansk University of Technology, Poland

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 391 – 395

ISBN 978-83-60810-14-9
ISSN 1896-7094

# On stimulus for citizens' use of e-government services

Cene Bavec
University of Primorska, Faculty of
Management Koper, Cankarjeva 5,
6000 Koper, Slovenia
Email: cene.bavec@guest.arnes.si

***Abstract*—The paper presents the desk research on interdependences between individual use of e-government services and group of selected socio-economic indicators, results from public opinion polls on S&T, and work requirements in EU27 countries. We identified six distinct groups of indicators that are significantly correlated with use of e-government services: national innovativeness and competitiveness, regular use of Internet, demanding and autonomous work, interest in innovations and S&T, data protection and security and personal trust. Among others, research opens questions about possible role of social capital in public acceptance of e-governments services. Deeper insight into interdependences between studied indicators also reveals that old EU15 and the new EU member states in some cases demonstrate different behavior patterns.**

## I. Introduction

SUCCESSFUL implementation of e-government projects depends on public acceptance of new services. Practical experiences and researches [1][2][3] confirm that users' acceptance is not granted per se. Public approval is quite often below what developers expected [4]. To understand what motivates citizens to use e-government services is equally relevant issue for policy-makers and for developers [5]. Public providers cannot "force" individual citizens to use their services, as governments or corporations can do with their own employees. Motivation of citizens is often underestimated and many governments believe that it is enough to passively offer new services. Providers of e-government services should be more aware also of users' absorption ability for new technologies and applications. Some categories of citizens like elderly population [6] and people with special needs are already studied, but we are referring to general population.

From this point of view, it is attention-grabbing to see that the use of e-government services strongly vary across different European countries [7][8] The usage is clearly correlated with economic power of particular country and its ability to invest into development of e-government. But, economy cannot explain all regional differences. There are also other forces that can influence use of public services [9][10].

We were particularly interested in the role of socio-economic environment and citizen's perception of new technologies, as unseen forces behind innovativeness and consequently use of e-services [11]. Many researches confirm that social capital and other social characteristics strongly influence individuals' behavior and make them more or less opened for new ideas [12][13]. We still lack comprehensive definitions of social capital and variables, so we are using many substitutes like general trust. We took similar approach in our research, including results from selected public opinion polls in EU27.

## II. Research and hypothesis

In the paper we present a part of a wider research on interdependences between socio-economic environment and national performance indicators. In the focus of this particular research was the question which social and citizens' characteristics identify an environment that is favorable for use of e-government services by individuals. We were also interested in regional differences between EU countries, partly because it was relevant issue for policy-makers, and partly because there were available data on EU level. Socio-economic diversity in the EU is very high and offers an opportunity to study its influence and interdependences between different national performance indicators. In our desk research we also searched for possible dissimilarity between old EU15 and new EU member states.

Research hypothesis were based on prevailing perception of users' behavior like: high national innovativeness and positive attitude towards science and technology, high public interest in Internet and new on-line technologies, trust and awareness on personal data protection they all stimulate use of e-government services. In our previous researches [1] we noticed that in many cases old and new EU member states behaved differently, so we were interested to see if they follow the same pattern or not.

Main information sources for research were EUROSTAT data bases and public opinion polls published in Eurobarometers (Table 2). Our research sample was set of EU27 member states. Research was conducted in three steps:

- With factor analysis we reduced number of variables, considering only variables that load on the first principal component associated with use of e-government services by individuals (we started with nearly 50 indicators and ended with 24);
- In the second step we calculated correlations between remaining variables and the use of e-government services by individuals;

- To get a deeper insight into the structure of some particularly interesting correlations we decided to use also graphical presentations to visualize behavior of individual countries and their eventual clustering.

## III. Presentation of results

Correlations between use of e-government services and the most relevant economic indicators have been already recognized and interpreted (Table 1). These correlations just confirm that economically more developed countries can invest relatively more into e-governments than others, and consequently increase its usage. It is evident that innovativeness and national competitiveness are the most prominent characteristics of environments with high use of e-government services. Other correlations are not that high, so there is still a room for other often hidden forces that influence e-governments.

Less recognized and studied are correlations between use of e-government services and social indicators or public opinion which determine socio-economic environment that that can also significantly influence behavior of individuals. The Table 2 presents correlations with such indicators and

Table I
Correlations between e-government use by individuals and selected national performance indicators for EU27 member states

| | Source of data | Correlation with e-government use by individuals |
|---|---|---|
| Innovativeness (SII) | European Innovation Scoreboard | 0,881 ** |
| National competitiveness | IMD World CompetitivenessYearbook | 0,848 ** |
| GDP per capita in PPP | Eurostat | 0,724 ** |
| Economic performance | IMD World CompetitivenessYearbook | 0,653 ** |
| Labor productivity | Eurostat | 0,651 ** |
| Spending on human resources | Eurostat | 0,519 ** |
| Science and technology graduates | Eurostat | 0,376 * |

** Correlation is significant at 0,01 level.
* Correlation is significant at 0,05 level.

Table 2
Correlations between e-government use by individuals and selected socio-economic indicators for EU27 member states

| | Source of data | Correlation with e-government use by individuals |
|---|---|---|
| Share of individuals recently used e-commerce | Eurostat | 0,901 ** |
| Share of individuals regularly using Internet | Eurostat | 0,892 ** |
| High individuals' level of computer skills | Eurostat | 0,860 ** |
| Personal Trust | Spec. Eurobarometer 223 | 0,848 ** |
| Work at home | Eurostat | 0,836 ** |
| My job allows me to take part in making decisions | Spec. Eurobarometer 273 | 0,807 ** |
| My job allows me to use my knowledge and skills | Spec. Eurobarometer 273 | 0,798 ** |
| My job requires me to keep learning new things | Spec. Eurobarometer 273 | 0,758 ** |
| Life-long learning | Eurostat | 0,734 ** |
| Interested in innovations and S&T | Spec. Eurobarometer 224 | 0,693 ** |
| Properly protecting private information | Flash Eurobarometer 225 | 0,670 ** |
| Transmitting your data over the Internet is sufficiently secure | Flash Eurobarometer 225 | 0,593 ** |
| Interested in economics and social sciences | Spec. Eurobarometer 273 | 0,560 ** |
| Interested in Internet | Spec. Eurobarometer 273 | 0,456 ** |
| Globalization is opportunity | Eurobarometer 63 | 0,433 * |
| Well informed about inventions and S&T | Spec. Eurobarometer 282 | 0,381 * |
| Interesting in IT news in media | Spec. Eurobarometer 282 | -0,282 |

** Correlation is significant at 0,01 level.
* Correlation is significant at 0,05 level.

results of public opinion polls that were selected with factor analysis out of nearly 50 subjective selected indicators. They load on the first principal component which is associated with use of e-government services.

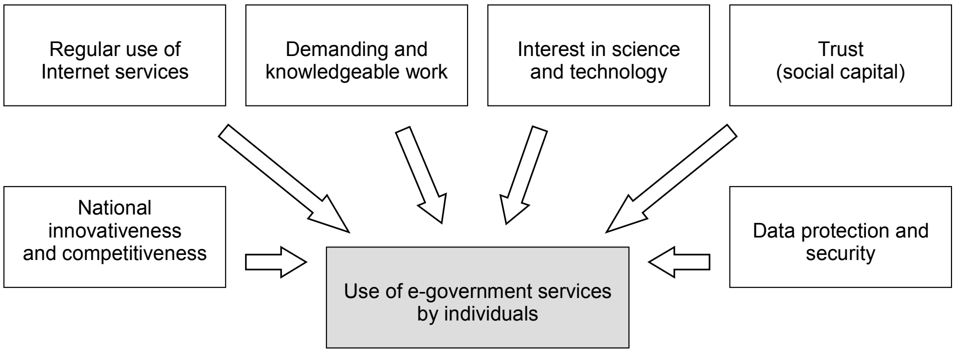

Regular use of Internet services
Demanding and knowledgeable work
Interest in science and technology
Trust (social capital)
National innovativeness and competitiveness
Data protection and security
Use of e-government services by individuals

Fig. 1 - Groups of indicators influencing use of e-government services by individuals

The first three top ranking indicators confirm high interdependence between use of internet services and use of e-government by individuals. It confirms that from user point of view e-government services are just ordinary e-services as others. More intriguing is high correlation with personal trust. It indicates a potential role of social capital (trust is an important part of social capital) that has not been really studied in connection with e-governments issues. It is also noteworthy that the next four places occupy indicators describing individuals' work conditions and professional demands. Extensive work at home and high demand for knowledge and personal initiative at work make individuals keener for use of e-government services. Similar effect has an intensive lifelong learning.

Public believe that personal information are properly protected is in the middle of the list with medium correlation (R=0,670). It is similar with public perception that transmitting data over Internet is sufficiently secure. It indicates that concern for data protection and security are relevant drivers for use of e-government services, but it is not decisive. Citizens tend to believe that their data will not be misused.

The next group of three indicators talks about individual awareness of S&T and Internet. Interest in S&T is much stronger motivator (R=0,67) than interest in Internet alone which demonstrate surprisingly low correlation (R=0,46). We can hypothesize that people accept Internet just as an useful tool, but they don't need to be very enthusiastic and interested in the tool itself. On the other side, general interest in S&T indicates a general innovative environment that is favorable also for implementation and use of innovative e-government services. Public perception that globalization is an opportunity is again a sign of open-minded society for new challenges. However, the correlation is already very low and it is statistically not really relevant.

Fig. 1 schematically presents six main clusters of national indicators and public opinions that are the most significant indication of stimulative environment for use of e-government services.

Correlations are just statistical figures and cannot reveal details in the structure of interdependences. For that reason we visually investigated positioning and eventual clustering of individual countries. As we already mentioned, the correlation between public interest in Internet and use of e-government services was low. This result contradict our common believe that sole interest in internet powers its use.

However, in the Fig. 2 we can notice two clusters of countries. In the first cluster are the most developed EU countries (Denmark, Netherlands, Sweden, Luxembourg, Finland, Germany, France and UK). They demonstrate high use of e-government services but a wide range of interests in internet, from very low to very high. We can conclude that these variables are nearly independent for this cluster. On the other side, in all other EU countries we can witness much lower use of e-government services, but much stronger correlation between it and interest in Internet. We can hypothesize that in the beginning of e-government implementation public interest in Internet plays a relevant role, but not in the more mature development phase when Internet becomes a "normal" and widely accepted technology. It indicates a nonlinearity that can be noticed in many other cases, too.

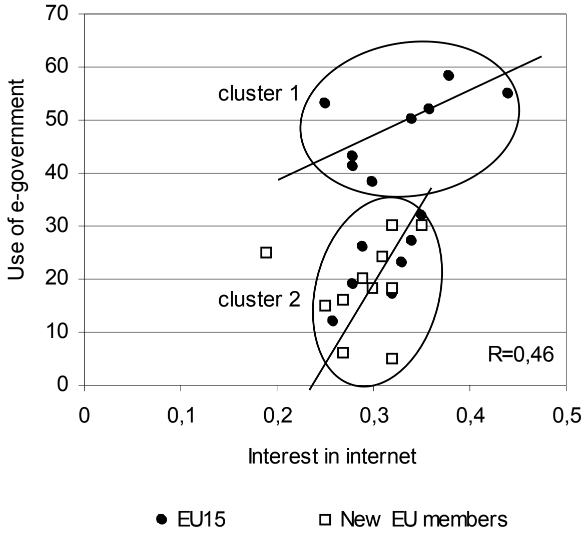


Fig. 2 - Interdependence between public interest in internet and use of e-government services

Stronger motivator for use of e-government services than interest in Internet is a general public interest in S&T (Fig. 3). In this case too, behavior of old (EU15) and new EU member states shows different pattern. In contrast to the old EU member states (full line in Fig. 3), in the new EU members higher interest in S&T doesn't result in considerably higher usage of e-government services (dotted line in Fig. 3). Relatively high interest in S&T in new EU member states is noticed in many researches and is attributed to educational system and some historical reasons. However, this interest alone cannot significantly raise use of e-government services because there are other, particularly economic brakes.

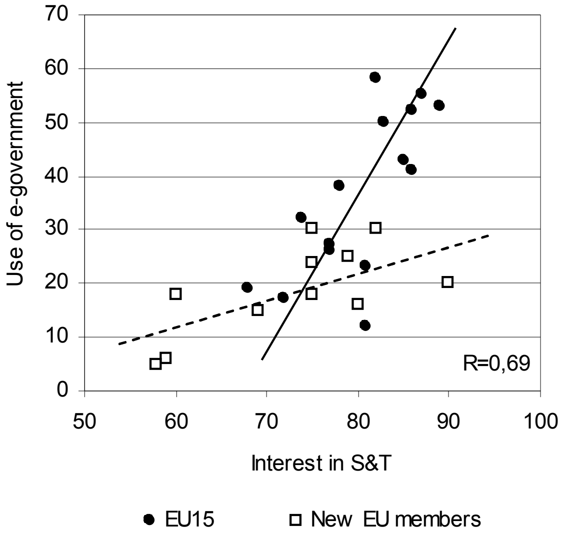


Fig. 3 - Interdependence between citizens' interest in S&T and use of e-government services

One of the strongest motivator for use of e-government services is a general national innovativeness. It identifies environment in which individuals are more attracted and opened to new ideas and ready to experiment with new technologies. So, it is not a surprise that such environment significantly stimulates use of e-governments. Investigating this interdependence in more details we can notice (Fig 4) that the most developed EU countries (marked cluster) have significantly higher level of innovativeness which is very likely one of the main reasons for their higher use of e-government services. However, we have to additionally comment this issue. The European Commission annually evaluates and ranks national innovativeness through Summary Innovation Index which also includes infrastructure issues and economic power that enable innovative processes in particular country. So, even national innovativeness is not just state of the mind, but reflects the strength of national economy. The circle is so closed, because the economic power makes possible higher investments into e-government projects and their higher use Table 1).

Correlation between the level of computer skills and use of e-government services by individuals is high (R=0,86), but again we can see two clusters of countries (Fig. 5). Trend lines are parallel but shifted indicating that at the same level of computer skills in northern EU countries exhibits higher level of e-government usage than in other countries that are on a lower "development" trajectory. In the new EU members, including Spain, Portugal, Italy and Greece, we see that even high level of computer skills results with lower use of e-government services. We can just guess that in the first phase of e-government development computer skill are more important and simulative than later stages.

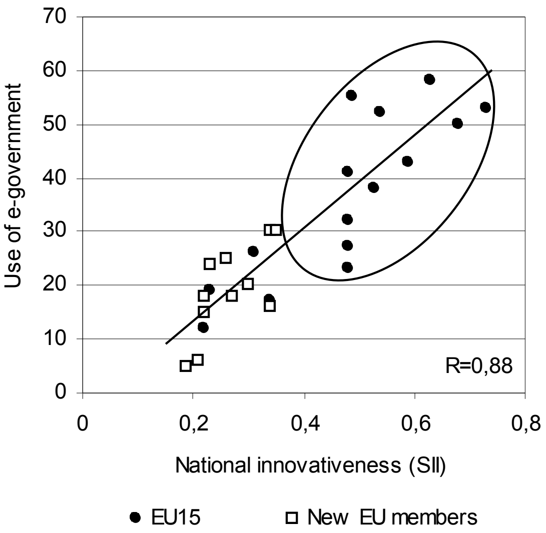


Fig. 4 - Interdependence between national innovativeness (Summary Innovation Index) and use of e-government services

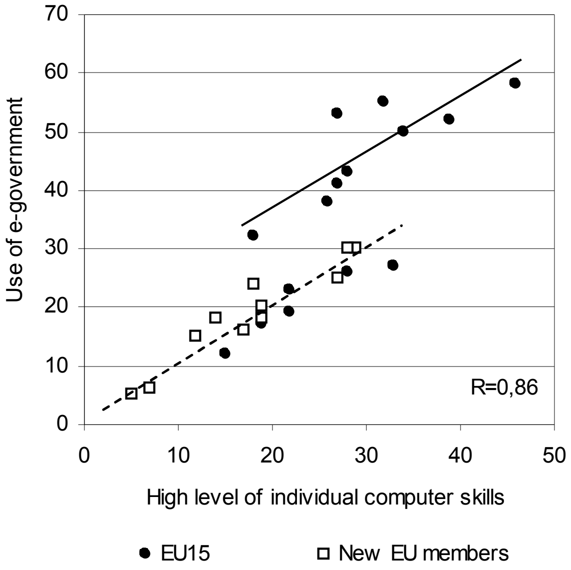


Fig. 5 - Interdependence between citizens' level of computer skills and use of e-government services

To summarize, figures from 2 to 5 illustrate four typical patterns in behavior dissimilarity in the old EU15 and new EU member states:

**P1:** In the most developed EU member states the use of e-government services is not very sensitive on changes in studied indicators (Fig. 2), but it is very sensitive in new EU members (and also for old EU15 members from the Southern Europe).

**P2:** The second pattern defines the opposite situation (Fig 3). In the new EU member states the use of e-government services is not very sensitive on changes in studied indicators.

**P3:** In the third pattern all EU member states demonstrates similar behavior (Fig. 4).

**P4:** The last pattern indicates different development trajectories for the most developed EU members and the rest, including all new member states and few old EU15 member states from Southern Europe (Fig. 5).

## IV. Conclusions

In the research we identified six distinct groups of national indicators and public opinions that are significantly correlated with use of e-government services by individuals. Two of them are quite evident (National innovativeness and competitiveness, and regular use of Internet), the next are data protection and security , and other three are less obvious (demanding and autonomous work, interest in innovations and S&T, and personal trust).

Particularly intriguing is high correlation between use of e-government and personal trust. It opens many questions about possible role of social capital in acceptance of new e-government services. As we already mentioned, this issue have not been really studied in connection with e-governments'. We could hypostatize that high social capital in Scandinavian countries and UK significantly increase use of publically available e-government services and that it is one of the main reasons for regional differences.

Another interesting feature is clustering of countries with different behavior. Deeper insight into interdependences between discussed indicators reveals that the old EU15 and new EU member states often demonstrate different behavior patterns. It would be interesting for some future research to find out if these differences are caused by nonlinearities in functional interdependences between use of e-government services by individuals and presented socio-economic indicators, or by other reasons. We indicated that in early development and usage phases of e-government services some correlations are different than in more mature phases. But, we could also offer another possible explanation. Lower level of e-government usage is characteristic of all new EU member states and some old EU15 member states from the Southern Europe, so we could hypothesize that their different attitude towards governments is relevant motivating factor in the use of e-government services. Particularly in Central European countries we could still see some public reluctance against government that has deep historical roots.

## References

[1] C. Bavec, “On the current environments for e-government development in the enlarged European Union. Information Polity, 2006, y. 11, no. 3/4, pp. 197-206.

[2] R. Heeks, “Causes of eGovernment Success and Failure: Factor Model”, eGovernment for Development. 2003, http://www.egov4dev.org

[3] J. K. Lee, H. R. Rao, “Risk of Terrorism, Trust in Government, and e-Government Services: An Exploratory Study of Citizens’ Intention to use e-Government Services in a Turbulent Environment”, Management Science and Systems, University at Buffalo, YCISS Working Paper Number 30, 2005

[4] A. Deursen, J. van Dijk· W. Ebbers “Why E-government Usage Lags Behind: Explaining the Gap Between Potential and Actual Usage of Electronic Public Services in the Netherlands”, Lecture Notes in Computer Science, Volume 4084/2006, pp. 269-280

[5] C. Centeno, R. van Bavel, J. C. Burgelman, “eGovernment in the EU in the next decade: Vision and key challenges”, European Commission. DG JRC Institute for Prospective Technological Studies. 2004

[6] C. W. Phang, J. Sutanto, A. Kankanhalli, L. Yan, B. C. Y. Tan, H. H. Teo, “Senior citizens’ acceptance of information systems: A study in the context of e-Government services”, Accepted for Publication in IEEE Transactions on Engineering Management http://www.comp.nus.edu.sg/~atreyi/papers/senior-egov.pdf

[7] J. Wei, “Global comparisons of e-government environments”. *Electronic Government*, Vol. 1. No. 3, 2003, pp. 229 – 252

[8] P. Wauters, G. Colclough, “Online Availability of Public Services: How Is Europe Progressing? Web Based Survey on Electronic Public Services. Report of the 6th Measurement. Capgemini Belgium. June (2006).

[9] C. Bavec, M. Vintar, “What matters in the development of the e-government in the EU?” *Lecture Notes in Computer Science* , 2007, no. 4656, pp. 424-435.

[10] S. K. Sharma, “Assessing e-government implementations”, *Electronic Government*, Vol. 1, No. 2. (2004), pp. 198-212.

[11] W. Van Oorschot, W. Arts, “The Social Capital of European Welfare States–The Crowding out Hypothesis Revisited”, *Journal of European Social Policy*, Issue 1, 2005.

[12] R. Florida, I. Tinagli, I. “Europe in the Creative Age”. Heinz School of Public Policy and Management at Carnegie Mellon University. 2004, http://www.demos.co.uk/files/EuropeintheCreativeAge2004.pdf

[13] C. Bavec, “Interdependence between social values and national performance indicators: the case of the enlarged European Union”. *Managing global transitions*, fall 2007, vol. 5, no. 2, pp. 213-228. http://www.fm-kp.si/zalozba/ISSN/1581-6311/5_213-228.pdf

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 397 – 401

ISBN 978-83-60810-14-9
ISSN 1896-7094

# e-collaboration Platform for the Development of Rural Areas and Enterprises

Grzego rz Kołaczek
Wrocław University of Technology, Wrocław, Poland
Email: grzegorz.kolaczek@pwr.wroc.pl

Adam Turowiec[1,2],
Dominik Kasprzak[1]
[1]ITTI Sp. z o.o., Poznań
[2]AMI@Work Family of Communities
Email: {adam.turowiec,dominik.kasprzak}@itti.com.pl

Witold Hołubowicz
Department of Applied Informatics
Adam Mickiewicz University,
Poznań, Poland
Email:holub@amu.edu.pl

***Abstract*—This paper presents the basic assumptions of the Collaboration@Rural project (C@R) supported by the EU's 6th Framework Programme for Research and Technological Development. Apart from discussing primary objectives of the project—focused on supporting the development of rural areas by providing network-based collaboration environment—it also shows basic assumptions of a 3-layer reference model serving as the foundation for C@R architecture. As an example of service rendered available within such network collaboration environment, the implementation of a notification service component is presented herein.**

## I. Introduction

Accounting for more than 90 percent of the EU's territory, rural areas are inhabited by almost 60 percent of its population. For this reason, the development of rural areas has long been one of the major priorities of EU's policy [5]. Despite that, many of those areas are still challenged by serious problems.

In the world of globalisation, dynamic competition and free-market economy, agriculture and forestry-related enterprises must continue to improve their competitiveness. What is a serious issue in this context, the average income per capita in rural areas is usually lower compared with cities, skills resources being lower as well, and the services sector—poorly developed. Nevertheless, rural areas have much to offer too. First and foremost, they are the source of basic raw materials. Because of the natural resources they are also a valuable place of rest and recreation. Many people are considering living or working in the country, as long as they can count on access to proper services and infrastructure (including ICT infrastructure). Bearing the above in mind, the objective of EU's policy relating to the development of rural areas is to overcome the challenges facing the population of such areas, and to utilise their potential [5, 6].

Chapter 2 of this paper presents a general concept behind the Collaboration@Rural project, together with its main objectives. Chapter 3 contains the characteristics of a layer-structured reference model of C@R architecture, while the next one describes the concept of implementing the e-mail based notification service within the framework of this project. The final chapter contains a summary and some information about the policies of further actions.

## II. Objectives of the Collaboration@Rural project

Collaboration@Rural[1] project (C@R) is a 3-year-long project carried out since September 2006 as part of the EU's 6th Framework Programme for Research and Technological Development. The project is being implemented by a multinational consortium comprising over 30 partners—universities and research centres, companies (both market leaders like Nokia or SAP, and representatives of the SME sector) and international organisations (including FAO and ESA). Poland is represented by Adam Mickiewicz University from Poznań.

The goal of this project is to accelerate the implementation of Collaborative Working Environments (CWE) in the context of sustainable development of rural areas. According to this concept, C@R encompasses a number of R&D actions (from analyses to validations), which have been grouped in three major interest areas, also defining the framework of technical solution—C@R service architecture (layers): Collaborative Core Services (CCS), Software Collaborative Tools (SCT) and Living Labs (see 1).

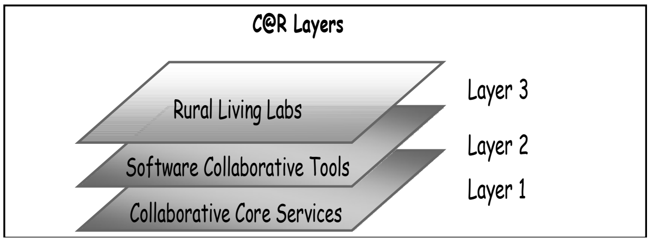


Fig. 1 C@R Architecture

C@R project implementation aims to complete five basic objectives, which have been defined in the following way [3]:

[1] Full title: *„A collaborative platform for working and living in rural areas*"

978-83-60810-14-9/08/$25.00 © 2008 IEEE

Obj 1: C@R will deliver collaboration environments to rural communities, to be defined in relation with the remaining network collaboration environments (CWE)

Obj 2: C@R will present the method in which three types of users can utilise the shared platform that integrates various hardware and software tools

Obj 3: C@R will promote the use of Open Collaborative Architecture (OCA) to cater for the needs of implementing new industrial and business projects in the rural sector, at the same time showing its usability and suitability for this type of actions

Obj 4: C@R will develop a concise methodology serving the development and assessment of obtained Rural Living Lab results

Obj 5: C@R will provide support to policy makers in the context of designing new strategies of development and innovation for rural areas after 2010.

## III. Layer-Based Reference Model for ICT Collaborative Environment

IT tools and technologies supporting group work and facilitating the creation of network collaboration environments (both for individual and group entities) have been the subject of studies, analyses and numerous research papers in the recent years. Owing to the continuous development and spread of technologies improving the efficiency of use of IT tools, it has lately become possible to carry out many specialist tasks (e.g. real-time positioning, teleconferences, telework, mobile workers' support, etc.) and reach communities that used to be marginalised (living in sparsely populated areas, rural areas, etc.)

The following 3-layer reference model has been approved for C@R, serving as the basis for the creation of an ICT collaboration environment for rural areas, comprising (see Fig. 2):

- Collaborative Core Services (CCS)—layer 1;
- Software Collaborative Tools (SCT)—layer 2;
- Rural Living Labs (RLL)—layer 3.

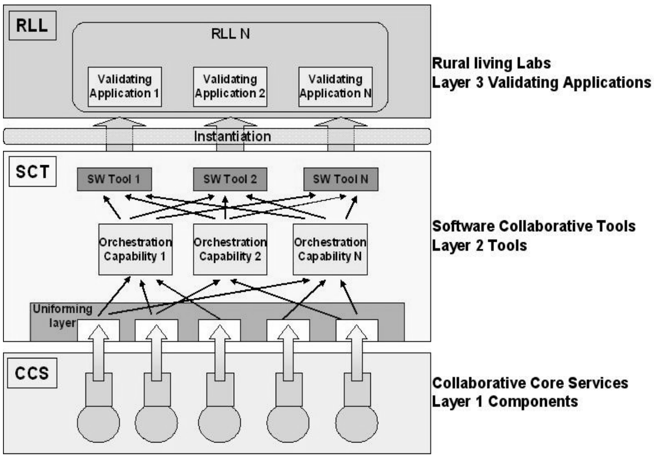


Fig. 2 Layered reference model for C@R [3]

The CCS layer includes software modules that allow to use all key platform services and resources (e.g. networks, sensors, devices, etc.). These modules can be freely combined with one another, and utilised with layer 2 SCT tools. With such flexibility, the C@R service architecture will be capable of cooperating with any solution and tool from outside the scope of the project (irrespective of its openness), which will allow the project to contribute significantly to defining the concept of Open Collaborative Architecture (OCA). The third layer—Living Labs—will render real environments available for the purposes of developing and validating solutions meeting specific needs of users from rural areas.

### *I. Collaborative Core Services Layer*

The layer of collaborative core services (CCS) is the first one in the 3-layer C@R reference model. It focuses on basic or low-level services and resources, which are indispensable for building a collaborative working environment (CWE). Consequently, elements of this layer comprise all mechanisms and technical means allowing to e.g.: provide access to ICT network, use advanced network services, render subject location service available, use the geographical information system, provide notification services and web services, deliver dispersed processing environment, etc.

All services available in this layer should be encapsulated in single programme components (CCS components) with specific objects defining data, protocols and API interface, so that they can form the basis for the structure of more complex services at the second layer of the model (comp. Fig.3).

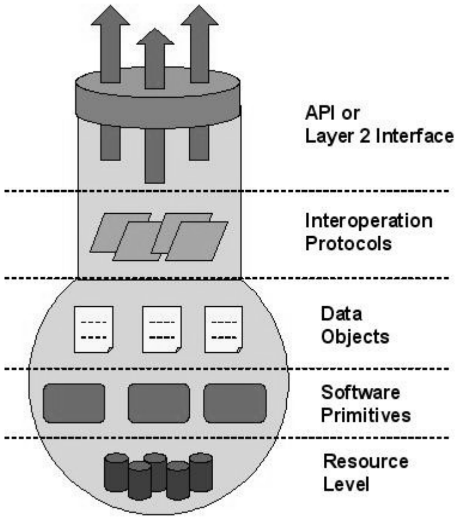


Fig. 3 CCS component construction [3]

The following CCS component classes have been defined in the C@R project:

- communications—comprising:
  - all possible network access systems (WiFi, WiMAX, UMTS, etc.);
  - advanced network services (QoS, multicast, etc.);
  - advanced services available via signal protocols (SIP/IMS, forwarding, multimodality, etc.);
- related to environmental and context data capture :
  - ICT infrastructure and devices (GSM/GPRS, GPS/Galieo, positioning services, terminals, etc.);

  - sensors and sensor networks providing data on environment variables, and allowing to identify subjects (e.g. data from RFID sensors required to identify goods, animals, etc.);
  - biometric devices and protocols (advanced methods of subject verification based on individual characteristics of eye retina, fingerprints, voice recognition, etc.);
- related to user experience :
  - all elements used directly by parties cooperating in rural areas, e.g. advanced graphic interfaces, computers built into tools, wardrobes, etc.;
- related to informati—on management:
  - all information sources;
  - data repositories—also dispersed ones;
  - data access technologies;
  - notification services;
  - web services.

### *II.Software Collaborative Tools Layer*

The second layer—Software Collaborative Tools (SCT)—is responsible for delivering the following three functionalities:

- Uniforming Middleware:

  it is a conceptual middle layer for CCS components, its goal being to harmonise, unify and adapt to existing standards; with this functionality, the C@R architecture gains flexibility and power, owing to improved integration with existing and newly developed standards;
- Orchestration Capabilities:

  this layer is responsible for delivering—within the network collaboration environment (CWE)—complex services, such as dispersed working environment, conditioning action and context; implementation of this objective consists in delivering mechanisms that allow to integrate elementary CCS components into more complex elements; key research issues relating to this functionality revolve around such elements as ontologies for collaboration environments, semantic compatibility, flows, synchronisation and coordination of middleware;
- Software Tools:

  they contain all necessary software components (scripts, programmes, intelligent agent programmes) in order to supply the end user with a component able to deliver specific services; basic components are related with synchronisation protocols, middleware orchestration protocols, dispersed repositories, context identification, multimodal interfaces and security.

### *III.Validation layer—Living Labs*

The concept behind a *Living Lab* is a methodology of conducting research and implementation activities in Knowledge-Based Economy and Information-Based Society conditions, where innovative products, services or applications are designed, tested and improved in real conditions, in interdisciplinary teams comprising all interested entities—from engineers and researchers, through entrepreneurs, local authorities and social organisations, to citizens.

A Living Lab is also a place (usually some area of a town, university campus, technology park, etc.) where innovation is developed—not merely in terms of technology, but also society, economy, etc.—focusing on the needs of the recipient. This unique anthropocentrism relating to the method of conducting research—i.e. focusing on people (citizens) as well as their requirements and expectations—is a characteristic feature of Living Labs methodology: man is the source of innovation here rather than the subject of testing, or the source of feedback needed to improve products. Moreover, the process of creating innovation is open [1,2] (i.e. based on a broad partnership of many different organisations) and democratic [4] (i.e. involving entire communities of end users). What is also important, each Living Lab is based on the collaboration of players of key importance to a given region (including local administration), owing to which it can engage in the completion of the strategic goals more efficiently.

The concept of Living Labs (LL) was developed within the framework of *AMI@Work Family of Communities* [7], the international expert group closely cooperating with the European Commission since 2004. Beginning from the Finnish Presidency (end of 2006) regions, cities and organisations which develop their own Living Labs (more than 50 currently) have been associated in the European ENoLL network (*European Network of Living Labs*). EnoLL, together with a number of projects co-financed by the European Commission (i.e. C@R accompanied by CoreLabs, CLOCK, COLLABS and others), have provided the background for creating numerous tools, guidelines and procedures aimed to facilitate developing and supporting the operation of Living Labs. Besides receiving support from the European Commission, this initiative has been supported by successive EU Presidencies and particular regions.

The role of Rural Living Labs (RLL) in the third layer of the C@R model consists in activating the Collaborative Working Environment—not merely in the final stage, being the validation of designed solutions, but also at the research level, e.g. when domain ontologies are being developed. C@R project entails continuous active involvement of RLL community members who are to be an important source of information about the accuracy of courses of research undertaken in each of the middle stages of the project, or allow to estimate the effectiveness of approved solutions at the level of individual system components. Another key role of RLL's is their social dimension, namely they allow to monitor social reception on an ongoing basis, along with the effectiveness of utilising proposed solutions.

Consequently, the third layer of this model is responsible for implementing the following tasks:

- causing a Collaborative Working Environment to acquire specific characteristics required by each of the RLL labs;
- selecting appropriate infrastructure or adapting existing infrastructure to each RLL, allowing to fully utilise the capabilities of lower layers;

- designing appropriate applications adjusted to specific needs and activities of each RLL;
- implementing a validation process based on action scenarios existing in each RLL, along with typical users' activity, in order to supply feedback to the designers of layer 1 and 2 components.

## IV. An Exemplary Implementation of the CCS Component—a Notification Service Based on Electronic Mail

One of the benefits notification services offer to entrepreneurs is the possibility of an immediate transfer of almost any information to a defined group of registered users. In particular, the entrepreneur or business partner is able to transfer their messages to cooperators and customers in the natural course. What is important is the fact that messages can be addressed to a selected group of recipients; thus, their content may be personalized and suited to the relation binding the entities, as well as to current needs.

Most entrepreneurs have already appreciated the benefits offered by this type of services and use them actively. Nevertheless, the fact remains that the solutions available today are characterized by a number of problems limiting the potential scope of their application. Among the commonly noticed drawbacks of the available solutions are: limited scanning possibilities, high dependence on the equipment and program platform, limited possibilities of configuration and management.

For these reasons, a CCS component allowing any entity reporting such functionality needs to benefit from the advantages of notification services is to be developed within the C@R project.

In order to demonstrate the possibilities of creating notification services integrated with the C@R platform, an SMTP-CCS component, allowing for sending e-mails, was developed. Its main element is the CCS, containing a Server capable of sending SMTP messages to an e-mail Server. The service provided by the SMTP-CCS component is available through the *Web Service* interface.

In order to take advantage of the possibilities offered by SMTP-CCS and the C@R platform, a client component needs to be implemented, which would enable access to the provided service.

### *IV.Resources included in the component*

The following resources were included in the components in order to create the notification service:

- SMTP-CCS contains a server capable of sending SMTP messages to an appropriate e-mail server; this server and the component are located in the same local network, thus facilitating communication. CCS makes the Web Service which allows for sending messages, available to the world (SendEmail);
- Client CCS contains a simple text interface enabling registration on the platform and the sending of basic e-mail messages with the use o the service provided by SMTP-CCS.

### *V. Component Specification*

- The server

The purpose of the SMTP-CCS component is to mediate between different C@R platform elements and the electronic mail server sending e-mail messages. The server is an element passively awaiting customer demands, which are serviced as they are sent.

The component implements the following functions:

- o Web Service client, which enables registration on the platform;
- o Web Service interface, which allows for obtaining connections with other elements of the platform;
- o Web Service interface which enables the reception of demands for a notification email.

- The client

The client component is an active element, which establishes connections with the server component for the purpose of sending notifications.

- o The component implements the following functions:
- o Web Sernice client which enables registration on the platform;
- o Web Sernice interface which allows for obtaining connections with other elements of the platform;
- o Web Sernice client which allows for sending connection demands to a different component;
- o A client servicing the data channel (it this particular case Web Service) which enables the sending of a notification through SMTP-CCS.

### *VI. The integration of the notification service with the C@R platform.*

In order to integrate the notification service with the platform, the following steps were taken:

The CCS core was fed with appropriate data enabling registration; “Notify e-mail” was indicated as the name of the provided service, being an available “WebService.SendEmail” data channel;

- The component’s Specific Options service was complemented with the necessary parameter of Web Service SendEmail address;
- The SendEmail service, which processes data, was implemented
- The SMTP-messages-generating functions were implemented;

In order to integrate the service client with the platform, the following steps were taken:

- CCS component framework was fed with the appropriate data enabling registration;
- The notification service search criteria were set for “e-mail”,
- A simple text interface allowing for testing client’s operations was implemented.

### *VII. Component interactions*

Figure 4. demonstrates the way in which components communicate. The sending of a notification entails realization of the four following stages:

1. The client component registers on the platform and sends an inquiry for availability of the server component (SMTP-CCS) and access parameters (protocol, address, etc.)
2. The client prepares a message including information describing the e-mail to be sent for SMTP-CSS. Once the message has been prepared, a suitable Web Service is called up (in this particular case SendEmail).
3. The SMTP-CCS server prepares and sends the SMTP message to the e-mail server.
4. The e-mail server is responsible for delivering the message to the addressee.

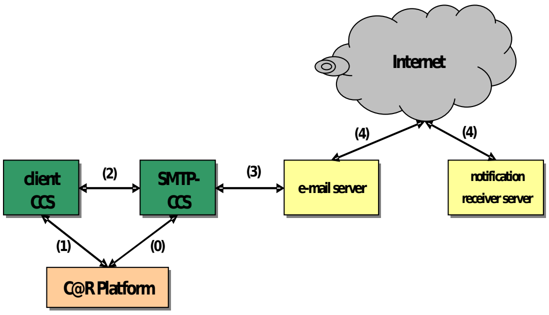


Fig. 4 Service components communications

Apart from the steps presented above, the SMTP-CSS registration on the platform (marked with "0") is required for the proper functioning of the service. This step enables other components to find the server and use the service it provides.

## V. Summary and further work

The article presented the IT platform of cooperation and its role in the process of rural areas development support. The basic assumptions of the Collaboration@Rural (C@R) project, implemented under the 6th Research and Technical Development Framework Programme of the European Union, were discussed as an important contribution to the support of stimulation of rural areas and preventing the phenomenon of digital exclusion. The three-layer reference model composed of the following layers: Collaborative Core Services (CCS), Software Collaborative Tools (SCT), and the so-called Living Labs was described in detail.

Moreover, a diagram of implementation of a selected component within the CCS layer was presented—a notification service using electronic mail.

Further works within the C@R project, which has entered into the second stage of its existence, will involve the validation of the results obtained within the CCS and SCT layers in the environment of RLL laboratories.

## References

[1] H. Chesbrough, *Open Innovation: The New Imperative For Creating And Profiting From Technology* , Harvard Business School Press, 2003.

[2] H. Chesbrough, *Open Business Models: How to Thrive in the New Innovation Landscape* , Harvard Business School Press, 2006.

[3] Collaboration@Rural: a collaborative platform for working and living in rural areas, Annex I—"Description of Work", 2006.

[4] E. von Hippel, *Democratizing Innovation* , Creative Commons 2005.

[5] Rural Development in the European Union—Statistical and Economic Information—Report 2007", OOPEC, 2007.

[6] The EU Rural Development Policy 2007-2013. Luxembourg: OOPEC, 2006.

[7] http://www.ami-communities.eu

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Corporate blogs—innovative communication tool or another internet hype? empirical research study

Grzegorz Mazurek, Ph.D.
Koźmiński University
ul. Jagiellońska 59, 03-301 Warsaw, Poland
e-mail: gmazurek@wspiz.edu.pl

***Abstract*—In the following paper the role, potential and perception of corporate blogging among key marketing decision makers from the companies listed on Warsaw Stock Exchange have been presented. The topic of blogs has been widely promoted in recent theoretical and practical publications, however very little information can be found on the scope of blogs usage and real impact of such modern communication tool on the business. Such issues as limitations in corporate blogs implementation, the perception of the blogs' information value or the expected potential of blogs usage have been identified and explained. Author describes various models of corporate blogs and tries to find out whether blogs are truly used as modern communication tool for business or it is just another hype which soon is replaced in media by another ideas and concepts.**

## I. Introduction

BLOGS, defined as web pages that serves as a publicly accessible personal journals [1] have been attracting media and public eye for the last few years. The amount of registered blogs is huge and still increasing - by the end of 2007, Technorati.com was tracking more than 107 million blogs [2]. Such phenomenon, appreciated and popularized by many internet users, has also been regarded in literature as innovative communication tool for business. That is why the new term has been coined – corporate blog.

A corporate blog is a weblog published and used by an organization to reach its organizational goals [3]. The purposes of blog usage can be grouped in three fields – brand building (incl. leadership), customer service (inc. product development) and promotion (incl. sponsorship and advertising) [4].

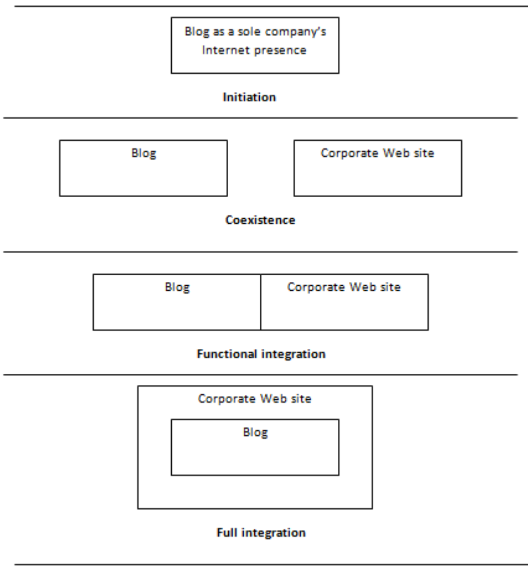
Corporate blogs have some unique features which make them perfect alterative or upgrade to typical corporate web pages which are usually exemplified by minor usage of user generated content (UGC) – i.e. the communication is one-sided or asymmetric and the users do not have many opportunities to provide company with valuable information, not mentioning about the possibility of having on-line dialogue with company's employees and other clients.

The key prerequisites for successful corporate blogs are [5]:

- symmetric communication (incl. using comments)
- informal language
- dialogue with readers which results in creating virtual community,
- regular postings,
- integration with other media and other content,
- clear rules and purpose of publishing (regulations).

Setting-up corporate blog often leads to challenges concerning the coordination of the overall on-line communication as companies use various other tools: corporate web pages, product or event sites and co-branded content via e-media presence. That is why 4 models of blog implementation are distinguished.

Scheme 1. Blogs and corporate web site – models of implementation



Source: [5], p.22.

The abovementioned characteristics of blog and media attention devoted to the issue should encourage companies to implement such innovative communication tool and widely popularize it among enterprises.

However, even rough market analysis proves that the usage of corporate blogs is very narrow – few companies in fact use corporate blogs. It shows that although blogs have unlimited opportunities and benefits, they have to face also many challenges as, for example: privacy and accountability issues [6], risk of loosing control over the communication strategy [7] or other issues raised in the presented empirical research.

Because corporate blogging seems to have huge potential in fulfilling various organizational roles, it is important to examine whether managers and specialists – taking into account their company's specific situation – perceive corporate blogs as tools they should use. That is why the second part of the article presents the corporate blog usage and its image among marketing decision makers in companies listed on the Warsaw Stock Exchange.

## II. Background—Blog Potential

There are different types of blogs. Dearstyne defines blogs on the basis of their source and impact and distinguishes two dimensions: external, internal [8]. Mazurek groups blogs on the basis of the three factors: function of blog, topic of blog and blog authorship [9]. In the following study the below mentioned corporate blog typology has been implemented:

- Leadership corporate blogs—in which particular person from company is thoroughly chosen to represent the company not only for brand building, but also for presenting company's views on various aspects. Such blogs are mainly used by companies which are market leaders.
- Group corporate blogs—in which all employees have right to participate in the development of company's blog. In such case, company uses only one blog, which is usually incorporated within the structure of company's corporate web site and the blog has many co-authors.
- Corporate blogs platform - consisting of many blogs written by employees or company's business partners. Such blogs can promote particular individuals as specialists in given fields and are often used as customer service support.
- Promotional blogs – such as sponsored blogs, advertising and contest blogs where the leading role is played by product, event or other marketing action, not by an author.

## III. Methodology

The findings presented below are taken from a survey conducted between May – 10 June, 2008 which was focused on the perception and usage of business blogs among marketing managers and specialists from the companies listed on the Warsaw Stock Exchange (GPW). Two research methods have been implemented in the survey. Firstly, corporate web sites of the listed companies have been examined in order to find out whether corporate blogs are used and what are their basic characteristics. Secondly, on-line and off-line questionnaire has been distributed among the marketing specialists and managers from the companies.

The overall research study was focused on five key areas:

1. Usage of corporate blogs—existence, function, model, aims
2. Reasons—why are the corporate blog used and why not
3. Perceived benefits—what are the top advantages of corporate blogs mentioned by those who don't use them yet
4. Blog information value—perceived value of information from corporate blogs for readers
5. Potential usage—whether respondents consider using corporate blogs in their companies.

## IV. Results

### *A.Population*

In the first part of research, 332 corporate web sites have been analyzed by experts which comprise ca. 98% of all listed companies. In the second part of research, based on on-line questionnaire supported by telephone and e-mail invitations, 57 managers and specialists from the listed companies responded to the survey. In addition, the off-line questionnaire has been answered by 39 managers and specialist from the total sample of 332 companies. In general, the response rate of answered questionnaire can be estimated on 29% of the total population (96 respondents).

### *B. Usage of corporate blogs*

All analyzed web sites were corporate sites, which means that the main aims they fulfill are: company brand building and information providing. Therefore, corporate blogs could have been perfectly implemented in such web pages in order to widen the scope and character of corporate communication with the environment. As the research shows, the companies using blogs as corporate communication tool are in vast minority. Only 17 (5% of the sample) of them use such tools and among them:

- 4 use leadership blogs,
- 4 use blog platform,
- 7 use promotional blog,
- 2 use corporate blog platform.

All the implemented blogs come from companies which deal with service sector. In particular, the blogs are published in companies from media, finance and insurance markets.

What is interesting, only few of the blogs noted significant comment publishing, most of them have had posts without any feedback from the readers—10 out of 17 identified corporate blogs have entries practically without any comments.

### *C. No usage of blogs—reasons*

Managers and marketing specialists from the listed companies are aware of corporate blogs potential (96% of respondents declares to know what corporate blogs are). Among the main reasons which were mentioned as important, very important or crucial in discouraging them from using blogs are:

1. Company's organization culture which doesn't accept such tools and informal way of communication ("closed companies")
2. Perceived problems with disclosure of important, secret information from the company to the public.
3. Lack of topics to write about which in consequence would lead to project failure.

The complete list of results are shown in Table I.

Among other reasons the respondents mentioned a few times were also: risk of spamming and flooding through comments function, black PR danger, no need as company uses other modern communication tools (discussion groups, chat rooms) and waiting for competitors and avoiding their mistakes afterwards.

TABLE I.
OPINION ON POTENTIAL CORPORATE BLOG BENEFITS

| Reason why to use blog in the future | 1 | 2 | 3 | 4 | 5 | Resp. | Total score | Average |
|---|---|---|---|---|---|---|---|---|
| 1. Improve basic e-marketing results (increased popularity among users) | 1 | 5 | 17 | 44 | 29 | 96 | 383 | 3,99 |
| 2. New, informal way of communicating with the environment (less formal) | 1 | 6 | 23 | 32 | 34 | 96 | 380 | 3,96 |
| 3. Create the leadership and innovative brand image | 3 | 10 | 41 | 23 | 19 | 96 | 333 | 3,47 |
| 4. Alternative way of customer service | 1 | 23 | 39 | 19 | 14 | 96 | 310 | 3,23 |
| 5. Get feedback from customers | 9 | 29 | 38 | 12 | 8 | 96 | 269 | 2,80 |
| 6. Promote the best employee and company's personalities | 1 | 48 | 29 | 11 | 7 | 96 | 263 | 2,74 |
| 7. Improve media relations | 10 | 33 | 39 | 5 | 9 | 96 | 258 | 2,69 |
| 8. Create virtual community | 1 | 53 | 28 | 8 | 6 | 96 | 253 | 2,64 |
| 9. Improve SEM position | 10 | 49 | 18 | 12 | 7 | 96 | 245 | 2,55 |
| 10. Sell products on-line | 41 | 29 | 16 | 7 | 3 | 96 | 190 | 1,98 |

Scale: 1 = Not a factor, 2 = Some, 3 = Important, 4 = Very important, 5 Primary reason

### *D. Perceived potential benefits*

Another issue raised by decision makers is the perceived advantages of corporate blogs usage. The results of the study shows that respondents considering corporate blog usage appreciate mainly the basic e-marketing benefits – increased traffic of users, information providing  and on-line brand image creating. Such issues as: another channel for customer service, valuable information feedback from the readers or improvement of media relations on-line are not widely appreciated, whereas – potential community building, search engine positioning and selling on-line are practically unnoticed by respondents.

The complete list of results are shown in Table II on the following page.

Among other potential benefits mentioned a few times by respondents we can find: employee integration, creating corporate identity among employees and reducing costs of traditional PR activities.

### *E. Information value*

The problem of reluctance in corporate blog implementation probably also comes from the managers' and specialists' opinion on the value of information they have from reading other blogs. 76% of respondents declare to read or scan through blogs and 45% of them are disappointed with the value them get from blogs. The respondents indicate that among the negative characteristics of blogs they have contact with, there are such disadvantages as:

- the entries are not interesting  (54%),
- the entries are irregularly updated (45%),
- the communication is one-sided (not comments or without comment option) (37%),
- the entries are heavily promotional (23%),
- the same texts can be found in other sources (12%).

In general, the perceived value of information gained from blogs the respondents read is estimated of 3.5 on 1-5 scale (1—not valuable, 5—very valuable).

On the other hand, perceived credibility of the massages presented on the read corporate blogs is estimated of 2,7 on 1-5 scale (1—not credible, 5—very credible). Such results prove that the blogs the respondents have contact with cannot be perfect examples of blog usage as they do not encourage the marketers to use the same tools in their companies.

### *F. The perspective of blogs usage*

The abovementioned results show that the image of blogs can hardly be called as very positive. However, more than 33% of respondents consider establishing corporate blog till the end of the coming year.

TABLE II.
DEFINING REASONS FOR NOT USING CORPORATE BLOGS

| Reason why not use blog | 1 | 2 | 3 | 4 | 5 | Resp. | Total score | Average |
|---|---|---|---|---|---|---|---|---|
| 1. Company's "closed" organization culture | 8 | 6 | 10 | 27 | 45 | 96 | 383 | 3,99 |
| 2. Perceived problems with disclosure of important information | 3 | 3 | 24 | 32 | 33 | 95 | 374 | 3,94 |
| 3. Prospective lack of topics to write about | 8 | 12 | 13 | 34 | 29 | 96 | 352 | 3,67 |
| 4. Employees reluctance to write | 2 | 8 | 38 | 25 | 23 | 96 | 347 | 3,61 |
| 5. Risk of receiving many negative comments and difficulty with dealing with them | 10 | 18 | 21 | 25 | 22 | 96 | 319 | 3,32 |
| 6. Having seen bad examples and users disappointment with blogs | 12 | 22 | 19 | 33 | 10 | 96 | 295 | 3,07 |
| 7. Risk of legal rights to text and other legal issues | 24 | 18 | 31 | 20 | 4 | 97 | 253 | 2,61 |
| 8. Lack of know-how in blog project management | 37 | 21 | 16 | 17 | 4 | 95 | 215 | 2,26 |
| 9. Risk of loosing valuable employees who promote themselves through blogs (head hunting) | 27 | 38 | 21 | 2 | 8 | 96 | 214 | 2,23 |
| 10. Budget constraints | 34 | 36 | 15 | 4 | 6 | 95 | 197 | 2,07 |

Scale: 1 = Not a factor, 2 = Some, 3 = Important, 4 = Very important, 5 Primary reason

50% of respondents who want to set up corporate blog in that time think about the promotional blog, 28% consider establishing group blog, 16% declare to create the blog platform for employees whereas only 6% think about the leadership blog.

On the other hand, more than 41% of respondents don't intend to implement any blogs in their e-marketing strategy.

In general, 57% of respondents agreed that the role of corporate blog usage will be increasing in the next years, 30% have opposite opinion, 13% couldn't say.

## V. Conclusion

The research study clearly illustrates that using corporate blogs can be described as being in the embryonic stage and the overall results – in the context of the advancement in corporate blog usage - are similar with the conclusions deriving from the analysis of corporate blogs in Fortune 500 companies where only 3.6% of companies used such tool in 2005 [10]. In 2008, among 332 companies listed on Warsaw Stock Exchange only 17 use corporate blogs (5% of the sample). The marketing managers and specialists from the researched companies very critically look at the real potential of corporate blogs for their specific situation – they acknowledge mainly brand building, web traffic improvement and information providing. Worth mentioning here is the fact that the most important reasons for postponing the corporate blog implementation in the listed companies were: the unfavorable organizational culture of company and perceived problems with disclosure of important information.

The respondents also declared that the value and credibility of information they receive from other blogs are not very high – such issues – combined with the others also mentioned in that study lead to the conclusion that the real image of blogs among marketing decision makers differ from the media hype around virtual diaries.

On the other hand, such critical view on blogs doesn't discourage the decision makers from using corporate blogs— still many of them consider using the tool—33% of respondents think about setting up corporate blog till the end of next year and 57% of respondents agreed that the role of corporate blog usage will be increasing in the next years

Those results are in fact encouraging as indicate that if corporate blogs emerge, they will by based on good situation analysis and critical view instead of short term fascination.

## References

[1] R. Blood, "The Weblog Handbook: Practical Advice on Creating and Maintaining Your Blog", Perseus Publishing, Cambridge, MA, 2002, pp. 12.

[2] http://www.technorati.com [access 12-06-2008]

[3] http://en.wikipedia.org/wiki/Corporate_blog [access 10-06-2008]

[4] J. Wright, "Blog marketing: The Revolutionary New Way to Increase Sales, Build Your Brand and Get Exceptional Results", McGraw Hill, USA, 2006, pp.13-20.

[5] G. Mazurek, "Blogi i wirtualne społeczności — wykorzystanie w marketingu", Wolters Kluwer, Kraków, 2008, pp.19-21.

[6] F. B. Viégas, "Bloggers' expectations of privacy and accountability: An initial survey". *Journal of Computer-Mediated Communication,* 10(3), 2005, article 12.

[7] D. Jones, "CEOs refused to get tangled up in messy blogs", *USA Today,* No.10 May, 2005.

[8] B. W. Dearstyne, "Blogs—the new information revolution?", *Information Management Journal,* Vol. 39 No.5, 2005, pp.38-44.

[9] G. Mazurek, "Blogi i wirtualne społeczności — wykorzystanie w marketingu", Wolters Kluwer, Kraków, 2008, p.22.

[10] S. Lee, T. Hwang, H. Lee, "Corporate blogging strategies of the Fortune 500 companies", Management Decision, Vol. 44 No.3, 2006, pp.316-34.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# An examination of factors affecting bidders' choice in electronic auctions

Costantinos Rougeris
Department of Business Administration
University of Patras
GR-265.00, Rio, Greece

George S. Androulakis
Department of Business Administration
University of Patras
GR-265.00, Rio, Greece
Email: gandroul@upatras.gr

***Abstract*—The increment of number of services provided in World Wide Web lately drives more and more consumers to e-commerce. During the last years and due to the vast increase of e-stores (B2C), electronic marketplaces and electronic auction sites became more popular. Many researchers examined how buyers interact with the auction facts and sellers during the procedure. Questions such as "how do millions of users decide about their e-bidding" and "what are the factors affecting them and what is their order of importance" are amongst the most significant ones in current research.**

**In this paper these factors are initially located using auction literature and eBay interface as well as expanded with the addition of a new factor (communication with seller). Their weights derive from the statistical analysis of the answers given in a questionnaire that was filled electronically by eBay users during the period of February – March 2007.**

## I. Introduction

E-AUCTIONING has raised some really interesting questions during the later years including the following:

- What are the unrelated factors affecting the eBay users to bid on an item?
- How important is each one of them for the decision maker?
- Do these factors have the same weight or does it change under different circumstances?

E-commerce latest standards instead of the traditional e-shops and on-line catalogs are e-auctions web sites. On-line auctions allowed a brand new model of data interchanging (C2C) to grow up rapidly during the last years. Consequently, a new theoretical and research field has been inaugurated, trying to interpret the physiology of on-line auctions.

Initially, there was an effort to describe e-auctions in comparison to traditional physical auctions and an analysis of the characteristics of the seller was also made [1], [2], [3], [4]. As the e-auctions evolvement continued, it was necessary that the factors affecting the buyer and their results being analyzed [5], [6], [7]. An interpretation of the buyer's behavior was also made based on those factors and on how can he or she decide for the same item between 2 auctions [8], [9].

In this paper e-auctions are examined from the buyer's view. All factors affecting the buyer that were examined previously are gathered and ranked by importance according to users' opinion. For the statistical facts, we are based on the results deriving from the comparison of 2 different questionnaires, one filled in September/October 2006 and the second filled in February/March 2007.

In section II the electronic auctions literature so far is presented while in section III the factors investigated in this paper are analyzed and how they affect the buyer to bid on an item. In sections V the results of the statistical analysis are presented and finally, conclusions and further research for future discussion are incorporated in section VI.

## II. Auction Theory

As Milgrom and Weber pointed, [9], introducing affiliated values (AVs), evaluation of an item under-auction is a result of objective but also personal factors. Especially when the item is bought not for resale but for personal and often collective purposes it is clearly a subjective issue, [3]. Therefore electronic auctions must be criticized not only on auction but on the bidder level too.

The various characteristics of on-line auctions and sellers investigated in [3], [10], where explained how they are converted into factors affecting the buyer in comparison to a physical auction.

Vakrat, [6], expanded the research on how much consumers were willing to pay for identical products offered through on-line auctions instead of on-line catalogs. The authors findings suggest that bidders prefer shorter auctions and expect larger discounts for expensive items, while their second study [7] concludes that most bids are made during the first half of the on-line auction, and that high starting bids result in fewer bidders and vice verse.

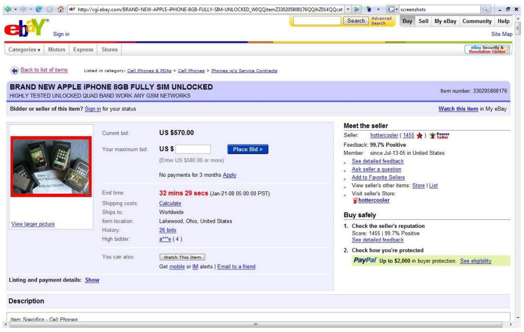
Bapna, [11], later categorized the bidders into types—evaluators, participators and opportunists—and describes their behavior during bidding.

Seller feedback rating has been widely discussed as a factor in many studies, [12], [10], [13], as well as long-time users usually bid on an item before the auction ends, [14].

It is also known, [8], that the starting bid of an auction, the number of negative comments and the length of the auction all affect the final price of the item.

In the same paper, [8], using data from 55,000 bids over a 3 years period, auctions of identical items that took place during different periods are examined. Kauffman and Wood found factors that affect bidders making them willing to pay more for the same item that are the existence of an image,

978-83-60810-14-9/08/$25.00 © 2008 IEEE


http://cgi.ebay.com/BRAND-NEW-APPLE-iPHONE-8GB-FULLY-SIM-UNLOCKED_W0QQitemZ330205808176QQihZ014QQcat
screenshots
Search
Advanced Search
Buy
Sell
My eBay
Community
Help
Sign in
Site Map
Categories
Motors
Express
Stores
eBay Security & Resolution Center
Place Bid >
Watch This Item

Back to list of items Listed in category: Cell Phones & PDAs > Cell Phones > Phones w/o Service Contracts

BRAND NEW APPLE iPHONE 8GB FULLY SIM UNLOCKED
HIGHLY TESTED UNLOCKED QUAD BAND WORK ANY GSM NETWORKS
Item number: 330205808176

Bidder or seller of this item? Sign in for your status
Watch this item in My eBay

Current bid: US $570.00
Your maximum bid: US $
(Enter US $580.00 or more)
No payments for 3 months Apply

End time: 32 mins 29 secs (Jan-21-08 05:00:00 PST)
Shipping costs: Calculate
Ships to: Worldwide
Item location: Lakewood, Ohio, United States
History: 26 bids
High bidder: a***e ( 4 )

You can also: Watch This Item
Get mobile or IM alerts | Email to a friend

View larger picture

Listing and payment details: Show

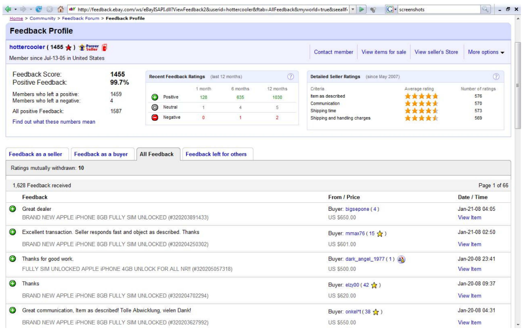
Meet the seller
Seller: hottercooler ( 1455 ★ ) Power Seller
Feedback: 99.7% Positive
Member: since Jul-13-05 in United States
- See detailed feedback
- Ask seller a question
- Add to Favorite Sellers
- View seller's other items: Store | List
- Visit seller's Store: hottercooler

Buy safely
1. Check the seller's reputation
Score: 1455 | 99.7% Positive
See detailed feedback
2. Check how you're protected
PayPal Up to $2,000 in buyer protection. See eligibility

Description

Item Specifics - Cell Phones

Fig. 1. The main eBay bidding screen. Time is ticking, all factors are presented and the item is waiting for yet another bidder.



Home > Community > Feedback Forum > Feedback Profile

Feedback Profile

hottercooler ( 1455 ★ ) Power Seller
Member since Jul-13-05 in United States

| | |
|---|---|
| Feedback Score: | 1455 |
| Positive Feedback: | 99.7% |
| Members who left a positive: | 1459 |
| Members who left a negative: | 4 |
| All positive Feedback: | 1587 |

Find out what these numbers mean

Recent Feedback Ratings (last 12 months)

| | 1 month | 6 months | 12 months |
|---|---|---|---|
| Positive | 128 | 635 | 1030 |
| Neutral | 1 | 4 | 5 |
| Negative | 0 | 1 | 2 |

Detailed Seller Ratings (since May 2007)

| Criteria | Average rating | Number of ratings |
|---|---|---|
| Item as described | ★★★★★ | 576 |
| Communication | ★★★★★ | 570 |
| Shipping time | ★★★★★ | 573 |
| Shipping and handling charges | ★★★★★ | 569 |

Ratings mutually withdrawn: 10

1,628 Feedback received Page 1 of 66

| Feedback | From / Price | Date / Time |
|---|---|---|
| Great dealer<br>BRAND NEW APPLE iPHONE 8GB FULLY SIM UNLOCKED (#320203891433) | Buyer: bigsepone ( 4 )<br>US $650.00 | Jan-21-08 04:05<br>View Item |
| Excellent transaction. Seller responds fast and object as described. Thanks<br>BRAND NEW APPLE iPHONE 8GB FULLY SIM UNLOCKED (#320204250302) | Buyer: mmax76 ( 15 ☆ )<br>US $601.00 | Jan-21-08 02:50<br>View Item |
| Thanks for good work.<br>FULLY SIM UNLOCKED APPLE iPHONE 4GB UNLOCK FOR ALL NR!! (#320205057318) | Buyer: dark_angel_1977 ( 1 )<br>US $500.00 | Jan-20-08 23:41<br>View Item |
| Thanks<br>BRAND NEW APPLE iPHONE 8GB FULLY SIM UNLOCKED (#320204702294) | Buyer: elzy00 ( 42 ☆ )<br>US $620.00 | Jan-20-08 09:37<br>View Item |
| Great communication, Item as described! Tolle Abwicklung, vielen Dank!<br>BRAND NEW APPLE iPHONE 8GB FULLY SIM UNLOCKED (#320203627992) | Buyer: onkel*t ( 38 ☆ )<br>US $550.00 | Jan-20-08 04:31<br>View Item |

Fig. 2. Bidders can read comments about all the seller's past transactions. They will give notice to the negative ones.

seller experience, whether or not the auction ends on weekend and previous bids (herd effect).

## III. Factors affecting an eBay auction

In this paragraph the most important factors affecting the buyers on an on-line eBay auction are introduced.

As presented in Fig. 1, a typical eBay bidding screen, bidders show all the information about the item and the seller in order to bid for an item. Items current price, time until auction ends (and the exact date and time it will end), seller feedback, description, item images, payment methods and other users bids are all there. Note that asking the seller a question about the product and the transaction is always an option for the bidder until the auction ends.

It is already known that the price of the item listed is the most important factor affecting an eBay auction. In this paper the desired price satisfying the bidder so as to buy online is to be located, expressed as a percentage of a physical store's price for the same item (discount). Moreover, we are trying to ascertain whether or not the bidder has a different discount requirement in comparison to the final price and also according to the item condition.

Time is also considered to be an important factor concerning eBay auctions, while it is connected with both price and the number of bids. More specifically, time is observed since the beginning of an auction and we are trying to locate the moment most buyers chose to bid.

In Fig. 2 a full seller's transaction history is presented and bidder is given the opportunity to read all the comments other bidders made about their purchases from the seller, positives or negatives. After each won item of an auction and the end of the transaction between buyer and seller, the former is obligated to leave a positive, negative or neutral comment (feedback) about the seller and the services he or she provided. The percentage of positive over negative comments is the seller feedback rating, which we is used to examine how it affects the buyer so as to bid for an auction.

Another recently introduced factor is how prior bids from other possible buyers on the same auction affect the bidder; thus it is interesting to find the bid ratio required to confirm our hypothesis. Supposing two auctions of the exact same item exist and share 10 bids, it is examined for what proportion will the buyer prefer one of them (i.e. 6 bids for the first auction over 4 for the second or 8 for the first over 2 bids for the second).

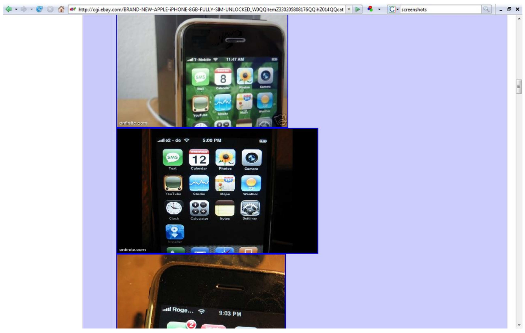
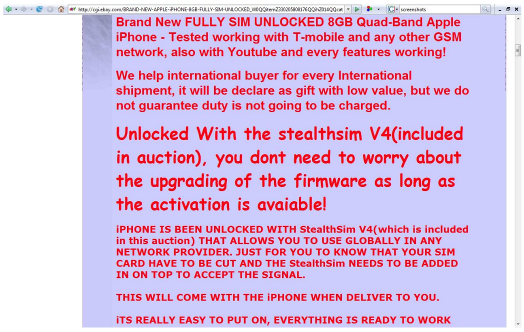

http://cgi.ebay.com/BRAND-NEW-APPLE-iPHONE-8GB-FULLY-SIM-UNLOCKED_W0QQitemZ330205808176QQihZ014QQcat
screenshots

Fig. 3. Several high resolution photos are needed between words.

In Fig. 3, a sample of large resolution images of the item used in the item's description can be observed. Note that images are "real" (of the specific item auctioned) and not taken from the item's manufacturer (i.e. from the company site). Image existence is also significant to the buyer's opinion on the item. The current findings on the topic is expanded by examining how it affects the buyer and whether or not more than one photo affects more. There is also a classification concerning the item condition (new item / used item in excellent condition / used item in good condition).

Except for the item's image existence and length, it is of equal importance to examine what is the item's description specific role in the bidders decision. Fig. 4 shows the importance of the description's extent in an auction. In this auction case, seller uses a large description of the item auctioned to persuade other users to bid on it, explaining the item's specifications and condition. It is examined (in relation to the item's condition: new item / used item in excellent condition / used item in good condition) whether and how a more detailed description in the item's presentation affects the buyer against a more concise one with less information about the item.

One factor that always influences eBay buyers is the way they have to pay for an item they bid on. The most popular way to pay for an eBay item is examined. It is noted that every auction on eBay can have one or more accepted payment methods that affect the bidder's choice.


http://cgi.ebay.com/BRAND-NEW-APPLE-iPHONE-8GB-FULLY-SIM-UNLOCKED_W0QQitemZ330205808176QQihZ014QQcat
screenshots

Brand New FULLY SIM UNLOCKED 8GB Quad-Band Apple iPhone - Tested working with T-mobile and any other GSM network, also with Youtube and every features working!

We help international buyer for every International shipment, it will be declare as gift with low value, but we do not guarantee duty is not going to be charged.

Unlocked With the stealthsim V4(included in auction), you dont need to worry about the upgrading of the firmware as long as the activation is avaiable!

iPHONE IS BEEN UNLOCKED WITH StealthSim V4(which is included in this auction) THAT ALLOWS YOU TO USE GLOBALLY IN ANY NETWORK PROVIDER. JUST FOR YOU TO KNOW THAT YOUR SIM CARD HAVE TO BE CUT AND THE StealthSim NEEDS TO BE ADDED IN ON TOP TO ACCEPT THE SIGNAL.

THIS WILL COME WITH THE iPHONE WHEN DELIVER TO YOU.

iTS REALLY EASY TO PUT ON, EVERYTHING IS READY TO WORK

Fig. 4. Description can be up to several pages, so as to persuade the bidder that "it's the real thing".

In holidays (non-working days) Web and thus eBay traffic is increased while users have more free time. It is attempted to investigate the effect of the phenomenon and whether or not the probability of a buyer to make o bid for an item auctioned on eBay is also increased on holidays.

Finally, another important factor affecting on-line auctions is the communication during the auction between the 2 parts (buyer and seller). The seller's ability to respond to the buyer's questions about the item auctioned affects the prospect of the latter to bid for it.

## IV. Methodology

This essay is based on answers collected by eBay users using a questionnaire and examines the factors that affect the bidder during his/her decision to bid on an eBay auction item. Although all the factors mentioned by previous authors are gathered, we again examine all points of an auction that could possibly irritate the bidder. We also rank them by levels of importance based on our results, being able finally to find how much does each of the factors affect the bidder on his decision. Whether the factors remain stable or have different weights according to the circumstances is also to be examined. A primary sample survey was created to evaluate the final questionnaire form; 21 questions were categorized in 12 categories investigating the factors affecting the bidder's decision. The questionnaire was anonymously filled out, while participating was completely volunteering. The type of questions used was multiple choice with one answer accepted and throughout the questionnaire clarity was the main goal, which is one of the most popular questionnaire issues. Answer-driven questions were also avoided, creating neutral and clear expressions. The questionnaire was available in two language versions (English and Greek), using the questionnaire developing software LimeSurvey, (www.limesurvey.org, 2007) and was hosted on a University of Patras server. The questionnaire is presented in Appendix A.

All 218 surveys were filled out between February and March 2007 by eBay users. The sample emanated from users that maintain accounts in various interest forums. Finally, for the statistical analysis of the questionnaire results was used the R-Project [15].

TABLE I
FACTORS' STATISTICAL MEASURES ACCORDING TO ITEM'S CONDITION.

| Category | Mean | Std | Skewness | Kurtosis |
|---|---|---|---|---|
| Item Price | | | | |
| $K_1$ | 2.624 | 0.6620 | 0.2050 | -0.4144 |
| $K_2$ | 3.234 | 0.6892 | -0.4222 | 0.5459 |
| $K_3$ | 3.569 | 0.6349 | -1.2794 | 0.9465 |
| Seller Feedback | | | | |
| | 3.023 | 1.0089 | -0.6352 | -0.7981 |
| Item Image | | | | |
| general | 3.362 | 0.6935 | -1.6184 | 2.5490 |
| $K_1$ | 3.479 | 1.5488 | -0.4451 | -1.3740 |
| $K_2$ | 4.561 | 0.9561 | -2.3700 | 4.8722 |
| $K_3$ | 4.706 | 0.8946 | -3.1528 | 8.9813 |
| Item Description | | | | |
| general | 4.280 | 0.9929 | -1.6184 | 2.5490 |
| $K_1$ | 3.688 | 1.3556 | -0.6743 | -0.8102 |
| $K_2$ | 4.390 | 0.9205 | -1.7233 | 2.8864 |
| $K_3$ | 4.541 | 0.9261 | -2.3723 | 5.3896 |
| Communication | | | | |
| | 4.518 | 0.8102 | -2.2330 | 6.0759 |
| Time until auction ends | | | | |
| | 3.908 | 1.3305 | -0.9194 | 0.4865 |
| Prior bids (herd effect) | | | | |
| | 2.683 | 1.3563 | 0.1304 | -1.2204 |

To evaluate the weight of each factor, every questionnaire participant was asked to rank the factors in decreasing order of importance. Furthermore, in order to discover whether or not more undiscovered factors affecting the the bidders' existence, the participants were also asked to and rank them among the others.

Another observed fact is that bidders change their stance on bidding depending on item's condition. For example, real images are strongly needed in cases of used items. In order to measure the changes in factors' weights a discrimination of 3 item condition categories was made:

$K_1$: new items
$K_2$: used items in excellent condition and
$K_3$: used items in good condition

## V. Survey Findings

In Table I all the factors investigated are listed with statistical results for each one of them. Factor results are discriminated when needed in our three condition categories ($K_1$, $K_2$ and $K_3$) and mean, standard deviation, skewness, kurtosis are shown based on our questionnaire answers.

A glance at the factors results can lead us to some early conclusions. First of all, according to the majority of the studies, price mean increases as the item condition deteriorate. The same increase appears also in item image and item description with the largest difference between new and used item (slightly or heavily used). This strongly endorses the bidder's need for large images and description of the item. Whatever the item's condition is, when it is not new, bidder has to be reassured that he knows what he is investing on. Concerning seller feedback, it is notable that there is no important disaggregation between highly feedback rated sellers (most ratings are over 95%) which might make a negative comment rating more usable for the bidders.

Essential in order to create a complete view of the factors' literature are the results of a question in which bidders were asked to rank the factors by importance. In Table I the results of this question are presented along with each factor's calculated weight. The sample of the survey was adequate to present a ranking between the factors. Nontrivial is the existence of another unidentified factor which always ranks last.

As expected, price is clearly the most important fact about bidding, while seller feedback is the secondimportant. Places three and four (the order is not statistically clarified) are occupied by item's image and description which could be also easily predicted. The surprise was that communication with seller during the auction ranked fifth or sixth along with payment method. This factor has not been examined in other research cases and seems to be quite important for the bidder. Time until auction ends and herd effect ranked seventh and eighth, while auction end day ranked ninth.

### A. Answer's facts

The survey results lead us to hypothesis testing in order to present more concrete conclusions. Therefore, for each important observation we introduced a null hypothesis in order to accept or reject:

$H_1$: the average discount in *item price* is the same in all 3 item categories.

$H_2$: users do not care about *seller feedback* rating.

$H_3$: one third of bids are placed during the last hour of the auctions (*time until auction ends* factor).

$H_4$: users are not affected by prior bids (*prior bids* -herd effect factor).

$H_5$: the number of images is the same in all 3 item categories (*item image* factor).

$H_6$: *Item description* is equally important in all 3 item categories.

$H_7$: users do not care about the length of the *item description*.

$H_8$: users do not prefer using PayPal for their payments (*payment method* factor).

$H_9$: credit card payment and bank transfer are not equally preferred (*payment method* factor).

$H_{10}$: users do not care about the day the auction ends (*auction end day* factor).

$H_{11}$: users do not care about *communication with seller* during the auction.

All factors' hypotheses were rejected. Accordingly, we evaluate each factor's importance, while significant remarks are presented:

**a) Price:** statistical analysis shows that buyers require different levels of discount according to the item condition. The negative sign of kurtosis for item categories $K_2$ and $K_3$ show that a deeper inquiry of item categories is required.

**b) Seller feedback:** the investigation of the factors shows that 70% of buyers require seller feedback rating of at least 95% to make a bid. The fact that very high reputation rating is required for buyers so as to bid might lead to the conclusion that negative comments will have a much more significant effect on buyers choice than the positive ones. Thus, a negative comment rating approach in seller's feedback might prove more appreciated by eBay users.

**c) Item image:** item image is concerned from all buyers as necessary on an on-line auction. It was also expected that the necessity of the image increases as the condition of the item deteriorates.

**d) Item description:** statistical analysis also shows that a small percentage of buyers (less than 8%) is not affected by the length and the level of detailed presentation in description, while most buyers seem increasingly interested as the item condition deteriorates. This seems to agree with the findings of Kauffman and Wood [8] (more information on item leads to higher prices).

**e) Communication with seller during the auction:** in communication things are clear, although a small percentage (less than 13,5%) is observed not being affected by this factor or slightly affected.

**f) Payment Method:** the payment method ranking was as expected. Most users seem to seek PayPal for their item e-payment, while second most popular method is credit card use. Bank transfer is way the third most popular method and personal cheque fourth which seems to serve very few.

**g) Time until auction ends:** according to the questionnaire answers, about 50% of the bids for an item are placed during the last hour of the auction, 25% are placed 1 to 3 hours before the auction ends, 12,5% are placed 3 to 12 hours before the auction ends etc. The findings show that "bid sniping" [14] is a strategy many users will choose to apply, whilst others won't tend to bid during the last three hours until their auctions end.

**h) Prior Bids (herd effect):** from the answers given we observe that eBay users are divided into two categories:

- in a small percentage (less than 18%) which is indifferent to this factor and
- in the majority (greater than 82%) of the users that take it under consideration. 57% of them seem to be decisively influenced by the herd effect.

Our findings here again agree with the conclusion of Kauffman's and Wood's [9] work about the prior bids in an auction.

**i) Auction end day:** about the day the auction ends, the larger percentage (not larger than 60%) remains indifferent by the day they will bid for an item, while the remaining percentage (not less than 40%) seems influenced. Differences in results may occur about this factor between studies as internet usage and weekend habits change from country to country.

**k) Other factor(s):** from the statistical analysis it is found that even if other factors affecting bidding exist, they fall short of importance on contrast to the others mentioned.

TABLE II
FACTORS' RANKINGS.

| Ranking | Factor | Factor Weight |
|---|---|---|
| 1 | Item Price | 0.8119 |
| 2 | Seller Feedback | 0.0640 |
| 3–4 | Item Image | 0.0294 |
| | Item Description | 0.0294 |
| 5–6 | Communication | 0.0215 |
| | Payment Method | 0.0215 |
| 7–8 | Time (until auction ends) | 0.0087 |
| | Prior bids (herd effect) | 0.0087 |
| 9 | Auction end day | 0.0043 |
| 10 | Other factor | 0.0006 |

## VI. Conclusions and future research

In this paper, the factors affecting a possible buyer in his/her decision to bid in an electronic auction were thoroughly examined. The most important findings of our work are:

(a) the classification of factors depending on their importance,
(b) the analysis of each factor as well as the appearance of a new factor (communication with seller during the auction).

More specifically, the importance of communication between buyer and seller was evaluated for the first time. It was also clarified that the existence of item image is essential for the user, as well as the extent of the description of the item. Seller feedback rating could be improved as an eBay feature, projecting negative comments in percentage of total. It would also be interesting to embed the item's price category to eBay's feedback system. Simultaneously, it was observed that although most users watch the item that interests them in-3-days time before the auction expires, they wait for up to the last hour in order to bid for it. Moreover, statistical analysis of all factors showed an explicit and measurable differentiation depending on the category of product (age). In addition to that, a small portion of buyers presents neutrality and/or indifference for certain factors.

In future research we will examine the reliability of our findings based on actual data drawn with special software from eBay or other online auctions websites. Another main goal is to improve the efficiency of our survey based on our current answers. For that, an additional and improved survey may be important. Finally, important research interest presents the application of modern techniques of data mining for the configuration and analysis of potential different profiles of possible buyers.

## References

[1] J. Y. Bakos, "Reducing buyer search costs: implications for electronic marketplaces," *Management Science*, vol. 43, pp. 1676–1692, 1997.

[2] ——, "Towards friction-free markets: the emerging role of electronic marketplaces on the internet," *Communications of the ACM*, vol. 41, pp. 35–42, 1998.

[3] R. A. Feldman and R. Mehra, "Auctions: theory and applications," *Staff Papers - International Monetary Fund*, vol. 40, no. 3, pp. 485–511, September 1993.

[4] E. Pincker, A. Seidmann, and Y. Vakrat, "Managing online auctions: current and business issues," *Management Science*, vol. 49, pp. 1457–1484, 2003.

[5] R. Bapna, "When snipers become predators: can mechanism design save on-line auctions?" *Communications of the ACM*, vol. 46, no. 12, pp. 152–158, 1998.

[6] Y. Vakrat and A. Seidmann, "Can on-line auctions beat on-line catalogs?" in *Proceedings of the 20th International Conference on Information Systems (ICIS 1999)*, P. De and J. DeGross, Eds. Charlotte, NC, 1999.

[7] ——, "Implications of the bidders? arrival process on the design of on-line auctions," in *Proceedings of the 33rd Hawaii International Conference on Systems Science (HICSS)*, R. Sprague, Ed. IEEE Computing Society Press, Los Alamitos, CA, 2000.

[8] R. J. Kauffman and C. A. Wood, "Doing their bidding: An empirical examination of factors that affect a buyer?s utility in internet auctions," *Information Technology and Management*, vol. 7, pp. 171–190, 2006.

[9] P. R. Milgrom and R. J. Weber, "A theory of auctions and competitive bidding," *Econometrica*, vol. 50, pp. 1089–1122, 1982.

[10] K.-H. Huarng and H.-Y. Cheng, "Online auction? a study of auction in yahoo! tawain," in *9th Joint Conference on Information Sciences*, ser. Advances in Intelligent Systems Research, H. Cheng, S. Chen, and R. Lin, Eds., no. article no 155. Atlantis Press, 2006.

[11] R. Bapna, P. Goes, and A. Gupta, "Insights and analyses of online auctions," *Communications of the ACM*, vol. 44, no. 11, pp. 42–50, 2001.

[12] S. Ba and P. A. Pavlou, "Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behaviour," *MIS Quarterly*, vol. 26, no. 3, pp. 243–268, 2002.

[13] T. T. Andrews and C. C. Benzing, "The determinants of price in internet auctions of used cars," *Atlantic Economic Journal*, vol. 35, no. 1, pp. 43–57, 2007.

[14] A. E. Roth and A. Ockenfels, "Last-minute bidding and the rules for ending second-price auctions: Evidence from ebay and amazon auctions on the internet," *The American Economic Review*, vol. 92, no. 4, pp. 1093–1103, September 2002.

[15] R Development Core Team, "R: A language and environment for statistical computing," 2006, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

## Appendix

### *A. Questions concerning the price of an item in auction.*

1) How lower should be the price of a brand new item be comparing to the price of a natural shop in order for you to bid for it on ebay.com?
   a) About the same price as the natural shop. (0–10% lower price)
   b) 10%-30% cheaper.
   c) 30%-50% cheaper.
   d) At least 50% cheaper (half price or lower).
2) How lower should be the price of a used item in excellent condition be comparing to the price of a natural shop's brand new one in order for you to bid for it on ebay.com?
   a) About the same price as the natural shop. (0–10% lower price)
   b) 10%-30% cheaper.
   c) 30%-50% cheaper.
   d) At least 50% cheaper (half price or lower).
3) How lower should be the price of a used item in good condition be comparing to the price of a natural shop's brand new one in order for you to bid for it on ebay.com?

a) About the same price as the natural shop. (0–10% lower price)
b) 10%-30% cheaper.
c) 30%-50% cheaper.
d) At least 50% cheaper (half price or lower).

### *B. Questions concerning the time until the auction ends*

4) In which exact time point during an auction would you bid for a product?
   a) Between 5 days and 24 hours until the auction ends.
   b) Between 24 and 12 hours until the auction ends.
   c) Between 12 and 3 hours until the auction ends.
   d) Between 3 and 1 hour until the auction ends.
   e) 1 hour or less until the auction ends.

### *C. Questions concerning the seller's reliability*

5) Which is the least acceptable seller's feedback rating in order for you to bid for a product in an auction on ebay.com?
   a) Not less than 90%-95% positive feedback rating.
   b) Not less than 95%-98% positive feedback rating.
   c) Not less than 98%-100% positive feedback rating.
   d) I don't care

### *D. Questions concerning the influence of prior bids in an auction in ebay.com.*

6) Please grade from 1 to 5 how important is for you the existence of prior bids from other possible buyers in an auction for a product in ebay.com:

   □1 □2 □3 □4 □5

7) Assume that there are 2 similar auctions of the same product with the same buying conditions. You have to decide on which one to bid. What should the adequate proportion between the prior bids be so as to be driven directly to bid in one of the auctions? (We assume that the prior bids are 10 in both the auctions) *Example: I would bid for the item that has at least 7 bids against 3 (meaning the one auction has 7 prior bids while the other has only 3, so I choose the first one)*
   a) 6–4
   b) 8–3
   c) 8–2
   d) 9–1
   e) 10–0

### *E. Questions concerning the influence of images during the auction.*

8) How many images in the item description do you consider as necessary enough for you to bid during an auction?
   a) None. The image is not a necessity for me to bid for an item.
   b) 1 image is enough.
   c) 2 images, 1 in the short and 1 in the extended description of the item.
   d) More than 2 images of the item.

9) Depending on the item condition how important is the existence of an image during an auction on ebay.com?

   For a brand new item.
   □1 □2 □3 □4 □5
   For a used item in excellent condition.
   □1 □2 □3 □4 □5
   For a used item in good condition.
   □1 □2 □3 □4 □5

### *F. Questions concerning the item description*

10) How much does a more detailed description of an item affect you in comparison to a shorter one during an auction in ebay.com?

   □1 □2 □3 □4 □5

11) Rate how much does a more detailed description of an item affect you in comparison to a shorter one during an auction in ebay.com depending on the item's condition:

   For a brand new item.
   □1 □2 □3 □4 □5
   For a used item in excellent condition.
   □1 □2 □3 □4 □5
   For a used item in good condition.
   □1 □2 □3 □4 □5

### *G. Questions concerning the Payment Method*

12) Which is your preferred payment methods in an ebay.com auction?
   a) PayPal
   b) Credit Card
   c) Bank Deposit / Transfer
   d) Personal Cheque

### *H. Questions concerning the bidding day*

13) During the non-working days (weekend-holidays) you more likely to bid on an item than the usual workdays?
   a) Not that I have observed.
   b) Maybe I little more like 0%-20% due to free time.
   c) During the holidays I use ebay.com 20%-50% more than the usual working days.
   d) I use Internet only on holidays so there is a 50% or more chance that I bid on an item these days than the usual working days.

### *I. Communication during the auction.*

14) How important do you consider the communication (quick and comprehensive answers about the price, shipping cost, item condition etc) between you and the seller before you bid for an item on ebay.com?

   □1 □2 □3 □4 □5

### *J. Other factors.*

15) Please describe any other factor not mentioned in current survey that could affect your choice for a bid in an auction on ebay.com

*K. Importance ranking of factors examined.*

16) Please rank the factors examined in the current survey according to each ones' importance. *(Please number each box in order of preference from 1 to 10)*

□ Item Price (including shipping costs).
□ Existence of one or more images in the description.
□ Time until the auction ends.
□ Positive seller rating.
□ Communication between buyer and seller.
□ Item description.
□ Payment method.
□ Bidding day.
□ Prior bids on an auction.
□ Other. (referring to question 15)

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Integration of governmental services in semantically described processes in the Access-eGov system

Marek Skokan, Tomáš Sabol
Technical University of Košice,
Faculty of Economics, Letná 9,
042 00 Košice, Slovakia
Email: {marek.skokan, tomas.sabol}@tuke.sk

Marián Mach
Technical University of Košice,
Faculty of Electrical Engineering
and Informatics, Letná 9, 042 00
Košice
Email: marian.mach@tuke.sk

Karol Furdík
InterSoft, a.s. Floriánska 19, 040
01 Košice
Email: karol.furdik@intersoft.sk

***Abstract*—This paper describes a "user-centred" approach to integration of services provided by the government in a "traditional" (i.e. face-to-face) or electronic way, applied in the EU R&D project Access-eGov. Individual ("atomic") services are integrated on semantic level (using semantic description of existing services – either traditional or e-services) into a scenario, realization of which leads to a solution of a problem faced by the end users (citizens or businesses) in a given life event (e.g. how to get a building permit, establish company etc.). Benefits for the end users are twofold: firstly they are provided with higher value-added services – a scenario of services (consisting of a series of services, including their dependencies), not just a single services; the services in the scenario are personalised (i.e. adapted to his/her personal data and/or situation). In case some services in the scenario are available electronically, they can be also executed online, increasing thus ultimate benefits to the end user. First prototype of the Access-eGov platform was test and evaluated in three pilot applications in three EU countries.**

## I. Introduction

INTEROPERABILITY was recognised as a precondition for the implementation of European eGovernment services in the eEurope Action Plan [1] and is explicitly addressed as one of the four main challenges in the i2010 EU strategy [2]. One of the most promising approaches to the interoperability is the employment of semantic technologies [3], [4]. Main advantage of this approach is the capability to formally describe meaning and context of government services, both traditional (i.e. face-to-face, "paper-based") as well as electronic ones (provided as electronic forms or web services), without necessity to modify the services themselves. The Access-eGov project (www.access-egov.org) builds on the use of semantic technologies with the aim to enable semantic discovery and semantic integration of governmental services into user specific scenarios. Integration of services into (user specific) scenarios (and the interoperability among them) is based on the WSMO technology (www.wsmo.org) used for description of process models by means of concepts defined in a knowledge model (ontology).

Access-eGov is a R&D project funded by the European Commission within the 6 th Framework Programme (FP6), Information Society Technologies (IST) programme. Within the Access-eGov project a SW platform supporting semantic interoperability of traditional as well as electronic government services in practical applications, together with methodological guidelines for introduction and management of such a platform, are being developed. In contrast to other projects, Access-eGov applies rather front office integration approach, i.e. no changes on the back office side are required.

The technological solution developed within the project is tested and evaluated within three pilot applications (in Slovakia, Poland and Germany) and one lab test (Egypt). The pilot application in Slovakia deals with the administration process of obtaining a building permission. The pilot application in Poland deals with the administration process of establishing an enterprise. The pilot application in Germany involves administrative and some non-administrative activities that are necessary to perform in a getting-married scenario.

## II. Semantic description of services

### *A. Web Service Modelling Ontology (WSMO)*

The WSMO framework (www.wsmo.org) provides a consistent conceptual model with the inclusion of mediators and distinction between goals and capabilities [9]. The Web Service Modelling Ontology (WSMO) is a conceptual model for describing semantic Web Services. WSMO consists of four major components: ontologies, goals, Web Services and mediators. Ontologies provide formal semantics to the information used by all other components. WSMO specifies the following constituents as part of the description of ontology: concepts, relations, functions, axioms, and instances of concepts and relations, as well as non-functional properties, imported ontologies, and used mediators. The latter allows the interconnection of different ontologies by using mediators that solve terminology mismatches.

A goal specifies objectives that a client might have when consulting a Web Service, i.e. functionalities that a Web Service should provide from the user perspective. In the WSMO, a goal is characterized by a set of non-functional properties, imported ontologies, used mediators, the requested capability and the requested interface.

A Web Service description in WSMO consists of five subcomponents: non-functional properties, imported ontologies, used mediators, a capability and interfaces. The capability of a Web Service defines its functionality in terms of preconditions, postconditions, assumptions and effects. A capability (therefore a Web Service) may be linked to certain goals that are solved by the Web Service via mediators. Preconditions,

978-83-60810-14-9/08/$25.00 © 2008 IEEE

assumptions, postconditions and effects are expressed through a set of axioms and a set of shared all-quantified variables. The service interfaces are described in the following chapter.

Mediators describe elements that aim to overcome structural, semantic or conceptual mismatches, which can appear between the different components that build up a WSMO description. Currently the specification covers four different types of mediators:

- OOMediators - import a target ontology into a source ontology by resolving all the representation mismatches between the source and the target;
- GGMediators - connect goals that are in a relation of refinement allowing the definition of sub-goal hierarchies and resolve mismatches between those;
- WGMediators - link a goal to a Web Service via its choreography interface meaning that the Web Service fulfils the goal; or link a Web Service to a goal via its orchestration interface meaning that the Web Service needs this goal to be resolved in order to meet the functionality;
- WWMediators - connect several Web Services for collaboration.

### *B. WSMO Choreography and Orchestration*

Interface of a Web Service provides further information on how the functionality of the Web Service is achieved. It describes the behaviour of the service from the client's point of view (service choreography) and how the overall functionality of the service is achieved in terms of cooperation with the other services (service orchestration).

A choreography description is semantically based on the Abstract State Machines (ASMs) [5] and consists of the states represented by ontology, and the if-then rules that specify (guarded) transitions between states. The ontology that represents the states provides the vocabulary of the transition rules and contains the set of instances that change their values from one state to another. The concepts of an ontology used for representing a state may have specified a grounding mechanism, which binds service description to a concrete message specification (e.g. WSDL).

For the Orchestration interfaces, it is planned by the WSMO authors to proceed as follows. The language will be based (note, that it is envisioned only, and the specification is not finished yet) on the same ASMs model as Choreography interfaces which - in order to link to externally called services or (sub)goals that the service needs to invoke to fulfil its capability - needs to be extended as follows:

- Goals and Services can be used in place of rules, with the intuitive meaning that the respective goal/service is executed in parallel to other rules in the orchestration
- The state signature defined in the choreography can be reused, i.e. external inputs and outputs of the service and the state of the choreography can be dereferenced also in the orchestration
- Additionally the state signature for the orchestration interface can extend the state signature of the choreography interface, with additional in/out/shared/controlled concepts which need to be tied to the used services and rules by mediators
- Respective WW or WG mediators need to be in place to map the in and out concepts defined in the orchestration to the respective out and in concepts of the choreography interfaces in the used services and goals, i.e. these mediators state which output concepts are equivalent to which input of the called service/goal and vice versa

### *C. Modifications in the Access-eGov project*

The life event approach [6] was adopted for modelling of government services, where the life event concept plays a central role – as a formal representation of user's point of view, his/her needs and requirements. Implementation of this approach resulted in the necessity to add the following top-level WSMO elements to the WSMO specification:

- Life Events – as formal models of user's needs, consisting from multiple goals and services organised into a generic scenario and expressed by orchestration construction consisting from shared variables (i.e. instances of concepts that are used within this life event) and transition rules that enable customisation of the generic scenario into a user specific scenario based on the user situation (i.e. instances describing this situation).
- Services as a generalisation of Web service concepts. This approach enables to describe both electronic and traditional government services by means of a service profile, containing functional and non-functional properties, capabilities, and interfaces. If there is no executable service available for a traditional service, the textual description of the required inputs (e.g. documents and forms, etc.) and requested actions (e.g. visit of a particular office) is specified as the non-functional property.

Requirement-driven approach [7] was developed within the Access-eGov project to guide semantic modelling and annotation (i.e. description of services by means of ontological models) of services provided by the government. While goals and life events are modelled in the ontologies (knowledge models) developed within this approach, the result of the annotation is a formalised WSML representation of the ontology containing all the definitions (concepts, classes) of services.

### *D. Process description in the Access-eGov*

The current WSMO specification for the process model based on the ASMs is, based on experience in the Access-eGov project, not structured in a way suitable for interaction with human actors, which is required for eGovernment applications especially those supporting also traditional services. For this reason, we have designed and implemented a workflow-based extension to the WSMO specification. Besides the objectives to guide citizens to achieve specific goals, and to coordinate activities performed by all actors - citizens, traditional public administration services and web services, the following facilities were identified as useful for a process model to provide support for modelling orchestrated scenarios:

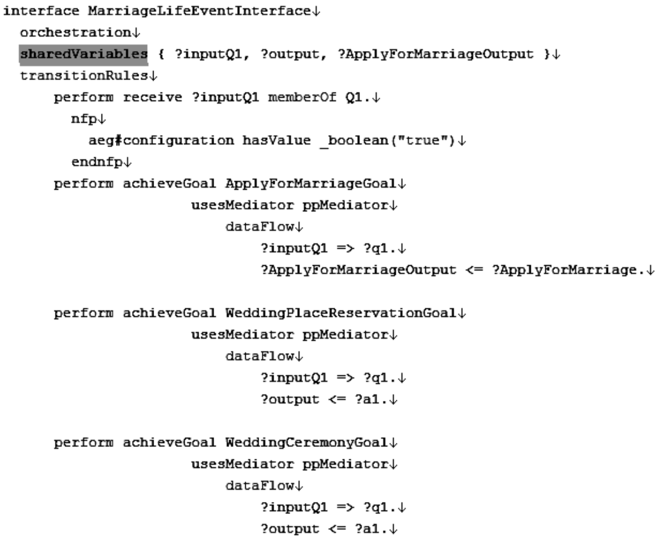
- compatibility with the standard process modelling notation (i.e. BPMN) in order to visualize scenarios to users and to use standard tools for modelling;
- compatibility with the proposed standard workflow modelling languages (i.e. WS-BPEL).

The Access-eGov model is based on the workflow CASheW-s model. The state signature is reused from the WSMO specification and replaces the ASMs transition rules with the workflow constructs. Shared ontology state signature allows reusing grounding of the input and output concepts to relevant communication protocols via WSDL for invocation of web services. Workflow model consists of activity nodes connected with the control or dataflow links. Each node can be either an atomic node (Send, Receive, AchieveGoal and InvokeService), or a control node (Decision, Fork and Join).

## III. Solution of life event

To put it simply, a WSML representation of a generic scenario is associated to the specified life event. This representation is then interpreted by the Acces-eGov system and presented to the user via SW tool called Personal Assistant client. The user answers relevant questions and if needed s/he chooses from a list of provided services.

The process of solving the life event situation consists of a set of specific goals that should be achieved, as well as from activities performed by all actors - citizens, traditional public administration services and web services. All these aspects are part of the process model (i.e. process ontology comprising generic scenarios) that is the core control (transition rules) and data (shared variables and data mediators) structure of the Access-eGov platform. Thus, process ontologies can be seen as an interface between the technical infrastructure design and the pilot applications. They provide a specification of the inner data structure for system components responsible for discovery, composition, mediation, and execution of services [8].



```
interface MarriageLifeEventInterface↓
  orchestration↓
  sharedVariables { ?inputQ1, ?output, ?ApplyForMarriageOutput }↓
  transitionRules↓
      perform receive ?inputQ1 memberOf Q1.↓
        nfp↓
          aeg#configuration hasValue _boolean("true")↓
        endnfp↓
      perform achieveGoal ApplyForMarriageGoal↓
                      usesMediator ppMediator↓
                          dataFlow↓
                              ?inputQ1 => ?q1.↓
                              ?ApplyForMarriageOutput <= ?ApplyForMarriage.↓

      perform achieveGoal WeddingPlaceReservationGoal↓
                      usesMediator ppMediator↓
                          dataFlow↓
                              ?inputQ1 => ?q1.↓
                              ?output <= ?a1.↓

      perform achieveGoal WeddingCeremonyGoal↓
                      usesMediator ppMediator↓
                          dataFlow↓
                              ?inputQ1 => ?q1.↓
                              ?output <= ?a1.↓
```

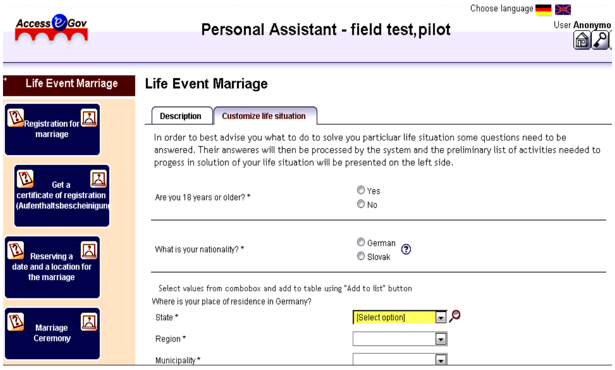
Fig. 1 Fragment of the process ontology of the pilot application of marriage in the new Access-eGov WSML notation

The above-presented figure is the high level process description of the life event getting marriage in Germany. The interface (MarriageLifeEventInterface) consists of two parts: sharedVariables and transitionRules. The first part defines variables (i.e. instances of concepts) that are visible within the whole interface of goal. Second part defines the process itself by using constructs from the set of the following constructs: if-then-enfIf, achieveGoal, send, receive. Note, that the example above uses only achieveGoal construct since it is high level process model (kind of complex goal) that is decomposed into three sub-processes (sub-goals).

In the Access-eGov syntax for process description the construct if-then-endIf is branching rule. This rule is used when we need to decide whether some constructs will be executed or not. The decision is done based on the evaluation of condition in the form of logical expression written in WSML syntax. When goal need to be decomposed into sub-goals the construct achieveGoal is used to address one of such sub-goal. There are three sub-goals of the presented goal (life event) marriage (ApplyForMarriageGoal, WeddingPlaceReservationGoal, WeddingCeremonyGoal). As can be seen the variables are mediated between goal and its sub-goals (construct usesMediator). The operator '=>' means that variable known in goal (that is on the left side of the operator) is known in sub-goal as variable on the right side of the operator (i.e. it's data mediation form goal to sub-goal). The operator '<=' means that variable known in goal (on the left side) holds data from variable known in sub-goal (on the right side) – i.e. data mediation form sub-goal to goal. The construct send means that process sends instance of specific concept to user. The construct receive means that the process needs instance of specific concept from the user side.

### *A. Presentation and interpretation*

The fragment presented above defines a part of the application that is presented to the user via the Personal Assistant Client. A screenshot of the Personal Assistant Client is depicted in Figure 2.



Access eGov
Choose language
User Anonymo
Life Event Marriage
Registration for marriage
Get a certificate of registration (Aufenthaltsbescheinigun
Reserving a date and a location for the marriage
Marriage Ceremony
Description
Customize life situation

Personal Assistant - field test,pilot

Life Event Marriage

In order to best advise you what to do to solve you particluar life situation some questions need to be answered. Their answeres will then be processed by the system and the preliminary list of activities needed to progess in solution of your life situation will be presented on the left side.

Are you 18 years or older? * ○ Yes ○ No

What is your nationality? * ○ German ○ Slovak

Select values from combobox and add to table using "Add to list" button

Where is your place of residence in Germany?

State * [Select option]

Region *

Municipality *

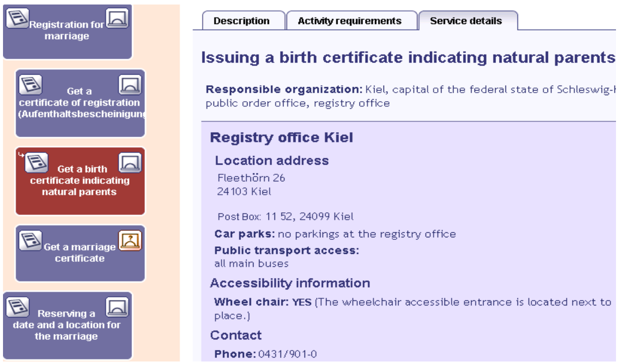
Fig. 2 Fragment of the German pilot application presented to users, that is defined in the presented Access-eGov WSML notation

Customisation of the user situation is based on the answers obtained from the user side that are internally (i.e. in the system) held as values within the instances of the concepts that are used in the process ontology. In this case the instance ?q1 of the concept Q1 holds answers to the questions about age, nationality and place of residence of the user. These answers are then used for the process customisation (i.e. insertion of sub-goal(s) or possibly for withdrawal of sub-goal(s) – not use in the current Access-eGov yet-) as well as for the service filtration.

Note, that on one hand those goals that do not have sub-goals (that cannot be decomposed) are considered as goals that might be resolved with the atomic administration service, or they represent complex part of the process that are not modelled. The latter means for the user that it is not possible to identify specific type of governmental services (and thus there are not instances of such kind of services described semantically in the system). Textual description of such goal is intended for navigation of the user. The example may occur in the German pilot application (getting married) in case that the spouse was born outside the EU and does not have German citizenship. Such case is not very generic and the set of required goals (most likely achievement of specific documents by using specific services) is not modelled. Currently, it's modelled just for cases when spouse is from EU in the Access-eGov system.

On the other hand those goals that contain sub-goals are considered as solved via services that resolve their sub-goals and with services that resolve them. In German pilot application, the example is goal 'Registration for marriage' that contains sub-goal 'Get a certificate of registration'. There is a specific service for both of these goals and the first goal is solved via the use of both of these services.

Note, that the existence of the service(s) to the goal (i.e. the existence of the specific kind of service that might be used by the specific user – e.g. in terms of place of user residence) is known to the user by the picture of office window drawn in the rectangle representing goal. A screenshot of the Personal Assistant Client with identified service is depicted in Figure 3.


Registration for marriage
Get a certificate of registration (Aufenthaltsbescheinigung
Get a birth certificate indicating natural parents
Get a marriage certificate
Reserving a date and a location for the marriage
Description
Activity requirements
Service details

Issuing a birth certificate indicating natural parents

**Responsible organization:** Kiel, capital of the federal state of Schleswig-H
public order office, registry office

Registry office Kiel

Location address

Fleethörn 26
24103 Kiel

Post Box: 11 52, 24099 Kiel

**Car parks:** no parkings at the registry office

**Public transport access:**
all main buses

Accessibility information

**Wheel chair: YES** (The wheelchair accessible entrance is located next to
place.)

Contact

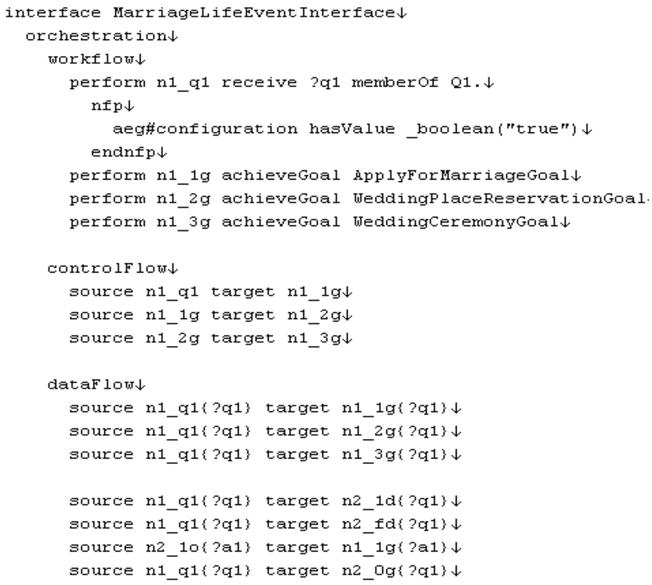
**Phone:** 0431/901-0

Fig. 3 Fragment of the German pilot application presenting service details of the identified service to the goal "Get a birth certificate"

Simply, after matching capabilities of the goal against capabilities of semantically described services the AeG system obtains service(s) that resolve goal. These capabilities are in the form of WSML logical expressions. The overall matching mechanism is not presented in this paper (more information can be found e.g. in [9]).

### *B. Practical experience with process description*

The evaluation of the first Access-eGov prototype was done within the first trial from October 2007 to the end of January 2008. Within this trial also quality of the ontologies and process models was evaluated. The results of this evaluation were analysed and implied changes that are currently being implemented. The second prototype will be tested and evaluated in the second trial in autumn 2008. The second prototype will incorporate also improved (easier) syntax for process description. Comparison of the old and new syntax is provided in Figure 1 (the fragment describes the life event getting married).



```
interface MarriageLifeEventInterface↓
  orchestration↓
    workflow↓
      perform n1_q1 receive ?q1 memberOf Q1.↓
        nfp↓
          aeg#configuration hasValue _boolean("true")↓
        endnfp↓
      perform n1_1g achieveGoal ApplyForMarriageGoal↓
      perform n1_2g achieveGoal WeddingPlaceReservationGoal·
      perform n1_3g achieveGoal WeddingCeremonyGoal↓

    controlFlow↓
      source n1_q1 target n1_1g↓
      source n1_1g target n1_2g↓
      source n1_2g target n1_3g↓

    dataFlow↓
      source n1_q1{?q1} target n1_1g{?q1}↓
      source n1_q1{?q1} target n1_2g{?q1}↓
      source n1_q1{?q1} target n1_3g{?q1}↓

      source n1_q1{?q1} target n2_1d{?q1}↓
      source n1_q1{?q1} target n2_fd{?q1}↓
      source n2_1o{?a1} target n1_1g{?a1}↓
      source n1_q1{?q1} target n2_0g{?q1}↓
```

Fig. 4 Fragment of the process ontology of the pilot application of marriage in the old Access-eGov WSML notation

The new syntax significantly simplifies the activity of process description. The most important positive aspect of the new syntax of process description is that it is not necessary to associate all usages of variables to the concrete node since the shared variables are known within the whole goal and the variables mediated between goal and its sub-goals are known within the whole sub-goals. This implies that identifiers of nodes are not needed. Another important aspect is that it is not necessary to define flow among the nodes. The process is read by the Access-eGov core system (execution mechanism) as a sequence as it is natural for the human reader too. Note, that the process description is done in the text based editor.

## IV. Conclusions

The Access-eGov system provides a consistent solution for description of processes within public administration, their interpretation and presentation to the user. This paper is focussed on the process description and some results of the first trial evaluation. The formalisms for the process description used in the Access-eGov project, represent an upgrade of the WSMO process description. This upgrade is based on the workflow CASheW-s model, therefore it is considered as compatible with the standard process modelling notation (i.e. BPMN) as well as compatible with the proposed standard workflow modelling languages (i.e. WS-BPEL). The first compatibility enables to visualize scenarios process models (scenarios) to the users and to use standard tools for modelling. Experiences gained so far within the Access-eGov show that the first version of the formalism proposed for the process description was difficult to read (understand) by public servants. For this reason, the syntax for process description was simplified, what will enable to check the correctness of the process description by public servants (i.e. not IT experts) and also to make corresponding changes (if needed). Thank to this, the administration of the Access-eGov system will be more flexible and easier, and the corresponding overheads lower.

## Acknowledgment

The Access-eGov project is co-funded by the European Commission within the contract No. FP6-2004-27020. The work presented in the paper was also supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 Project "Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge".

## References

[1] eEurope 2005: An information society for all. COM (2002) 263 final of 28 May 2002, http://europa.eu.int/information_society/eeurope/2005/ all_about/action_plan/index_en.htm

[2] i2010—A European Information Society for growth and employment, COM (2005) 229 final of 1 June 2005, http://ec.europa.eu/i2010

[3] Jochen Scholl, Ralf Klischewski, E-Government Integration and Interoperability: Framing the Research Agenda, in: *International Journal of Public Administration,* Vol. 30, Issue 8-9, 2007, pp. 889-920.

[4] A. Abecker, A. Sheth, G. Mentzas, L. Stojanovich (eds.), Proceedings of AAAI Spring Symposium "Semantic Web Meets eGovernment" (Stanford University, March 27-29, 2006), Technical Report SS-06-06, AAAI Press, Menlo Park, CA, 2006.

[5] Roman, D., Scicluna, J., Fensel, D., Polleres, A., de Bruijn, J.: D14: Ontology-based choreography of WSMO services. WSMO working draft, DERI (2006) Available online at http://www.wsmo.org/TR/d14/v0.4/ .

[6] Anamarija Leben, Mirko Vintar, Life-Event Approach: Comparison between Countries, in: *Electronic Government,* Springer LNCS 2739, 2003, pp. 434-437

[7] Klischewski, R., Ukena S.: Designing Semantic e-Government Services Driven by user Requirements. In: *Proceedings of ongoing research, project contributions and workshops, 6th International EGOV Conference,* September 3-6, 2007, Regensburg, Germany, pp. 133-140 (2007)

[8] Access-eGov Platform Architecture. Deliverable D3.1, Access-eGov Project, 2006. Available at http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_10_17_06_9_41_59_AM.pdf

[9] Marek Skokan, Peter Bednar, Semantic orchestration of services in eGovernment, in: V. Snášel (ed.), *Proc. of Znalosti (Knowledge)* 2008. STU, Bratislava, Slovakia, 2008, pp. 215-223

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 421 – 427

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Access-eGov–Personal Assistant of Public Services

Magdalena Sroga
Cities on Internet Association, ul. Prądnicka 4/15, 30-002 Kraków, Poland
Email: m.sroga@mwi.pl

***Abstract*—Aim of this article is to present targeted research project Access-eGov, founded from the Sixth Framework Programme , priority 2 - Information Society Technologies. The article describes Access-eGov architectural and logical design and used technologies which enable to achieve the Project's aims. The Project introduces new, user-centric approach which uses predefined life events to solve citizens' problems which require from users implementation of public services. Project's approach enables users to perform complex executive scenarios instead of separated public services. This approach concentrates on user's needs and helps to build semantic interoperability among e-government services.**

## I. Introduction

Access-eGov (Access to e-Government Services Employing Semantic Technologies) is the acronym of a project pursued by a consortium of European organizations. The project started in January 2006 and is going to be finished in December 2008. The aim of the Access-eGov results from the action plan eEurope which formulated specific aims in the area of eGovernment to make public administration open, transparent, inclusive and productive. In this area the specific objective of the Project is to support semantic interoperability among e-government services across organisational, regional and linguistic borders what is being achieved by employing semantic technologies.

In “real life” situations citizens, as well as businesses usually do not need an atomic (singular) government service, but a (often non-linear) sequence (including if-then-else branches) – it means “scenario” of atomic services. Since we are still far away from the situation where 100% of government services are available on-line (and the level of the availability of e-services varies quite significantly across the EU member states), users usually have to deal with a combination of traditional services and e-services. Due to that fact Access e-Gov’s approach considers both electronic and traditional services from which the user scenario (usually “hybrid” one) is composed. Before identifying and executing a scenario, users are sometimes facing a more trivial problem – which public administration institutions are providing service(s) which they need in the given situation (context) and what inputs are required to execute this service independently on the way of provision.

To solve the first users’ problem using Access-eGov system, currently existing public services may be annotated using Annotation Tool and afterwards registered in the Personal Assistant platform. All accessible services are enhanced by the detailed description, links to relevant documents and guidelines on service providers, thus the user is provided with the complete information related to the particular service. The second problem is solved by composing a user scenario from available services relevant to user’s life situation. A significant issue is that the user is provided with complete, customised information relevant to her life case. Customisation is achieved by employing semantic technologies, particularly WSMO ontologies.

## II. Background

In the last years a number of significant developments have occurred that motivate the use of Semantic Technology in eGovernment. It is well-known that Semantic Technology enables federation, aggregation and inferencing over information, as well as helps to solve interoperability, integration, reusability, and accountability issues in and across different institutions. There are created new systems supporting public administrations by Semantic Technologies and applications based on ontologies. But not many systems adopt the approach of user’s life events, which is going to be the core Access-eGov strategy – to simplify users’ interactions with public administration as much as possible.

Web Services are supposed to be the technology which supports realisation of cross-governmental, integrated services. They constitute the common technology to fulfil application-to-application interoperability, based on XML message exchange that is capable of dynamically invoking remote software components with a minimum effort in interface description and customizing.

In semantically-enriched systems, ontologies or controlled vocabularies are used as conceptual support for providing information about resources and for accessing them. Therefore public servants shall be empowered by Access e-Gov technology to annotate their institutions’ services on their own, being provided with intuitive software and straight-forward reference manuals.

Modern e-Government landscapes can be categorized in many ways according to solutions offered to their customers or according to their technical implementations. Criteria that Access-eGov takes into consideration are notably the easy accessibility of government services for the customers and the extent to which information systems can interact with each other in modern e-Government landscapes. The criteria of interoperability mainly involve dedicated interconnection

978-83-60810-14-9/08/$25.00 © 2008 IEEE

of information systems between several agencies on the same administrative level of government. Only in a few scenarios in the UK and Australia cross-governmental information links on a mutual basis and on different levels of government can be observed. Openness to external partners also includes the ability to interact on a technical layer with non-governmental or private organizations. Most advanced examples of modern service-oriented architectures and usage of message-based information exchange services in order to communicate between back-end-systems are: open service interfaces with OIOXML in Denmark, Public Service Infrastructure (PSi) in Singapore, Government eLink in Sweden, and Government Gateway in the UK.

The evolution towards integrated IT-based public services shows the necessity to adopt new ways of interaction between administration institutions. Some EU projects have been already developing practicable solutions to problems deriving from interoperability issues. They mainly focus upon semantic enrichment of electronic services and their aggregation and orchestration towards combined “complex e-services”. Nevertheless, most of these projects focuses on the technical side and still lacks a citizen-centred point of view that could be taken by implementing software components tailor-made for the citizen’s attendance when applying for a public service. In these projects, citizens' needs take a background position compared to technical aspects. Therefore, new approaches in e-Government have to put the emphasis on easy service accessibility for customers, what is the main difference between Access-eGov and those projects and also an noticeable value added of the project.

## III. Access-eGov Functionality and Components

An Access-eGov software component is any piece of software that performs a specific technical task – i.e. it does not fulfil a user requirement by itself, but it solves a low-level problem specific to a domain.

An Access-eGov software module is any piece of software that performs a specific functional task – i.e. a task that fulfils a certain user requirement, thus solving a high-level problem specific to the user domain.

Whereas a component cannot operate individually without the tasks performed by other software components, a module should be able to work independently of other modules and when the module stops working, other modules are not affected.

### *A. Access-eGov Logical Architectures*

The figure 1 illustrates general architecture of Access-eGov platform which may be sub-divided into the three major parts: the Access-eGov Infrastructure, the Access-eGov Personal Assistant client, and Access-eGov Annotation services (not an integral, but an affiliated part of the Access-eGov Infrastructure).

The services are owned by public administration and thus located on its premises. They are simply made available via the Access-eGov system and thus they are not an integral part of the overall system. These can be both electronic services (provided directly via web service interfaces or web forms) and traditional ones (i.e. provided face-to-face). Executable services will dispose of an electronic XML-interface to the Access-eGov Infrastructure, whereas traditional ones are only described and registered in the Access-eGov platform. They are supposed to be annotated by public agencies which are responsible for them and want to expose them to the public.

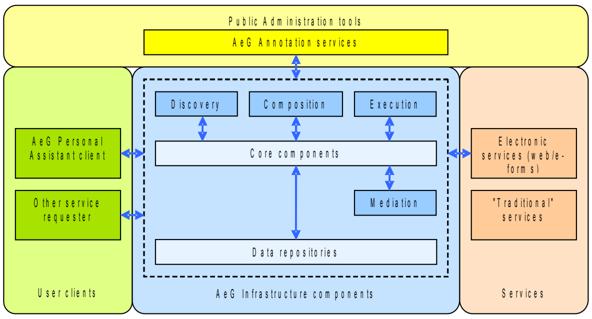


Fig 1: Overall architecture of the Access-eGov platform

Access-eGov Infrastructure components may be chosen by public agencies and installed on their premises or data centres dependently on which of them they wish to have. Such a “local” installation of the Access-eGov Infrastructure components is supposed to interact as a peer in the peer-to-peer overlay network that Access-eGov is likely to consist of.

### *B. Access-eGov Data View*

This section describes the relations between several data entities in order to provide a more detailed view on the platform data structures. The figure 2 presents structural correlations of data.

Life event denotes specific situation in user's life which requires performance of public services series. Life events can be categorized into groups and organized in multiple hierarchies. Using the Personal Assistant portal site, user may “browse” or navigate through life event categories in order to select an appropriate life event.

A life event may be assigned multiple goals, which will formalize user needs. Life event’s goals can have specified optional preconditions, which allow users to customize their specific life event. Preconditions are specified as logical expressions with input variables provided either explicitly by the user or from the user profile (preconditions are not dependent on service invocation).

Goal specifies objectives of the user who wants to perform a particular service, including functionalities that the service should provide from the user’s perspective. Goals formalize user needs by specifying the requested outputs and effects. This is declared in the same way as service functional properties.

Service profile specifies what the service does provide from user’s perspective and is used by the public administration to advertise services. Service profile consists of non-functional and functional properties.

Functional properties describe inputs, outputs, preconditions and effects of the service (IOPEs). They are specified as logical expressions, which consist of terms constraining type and property values of various resources required for or

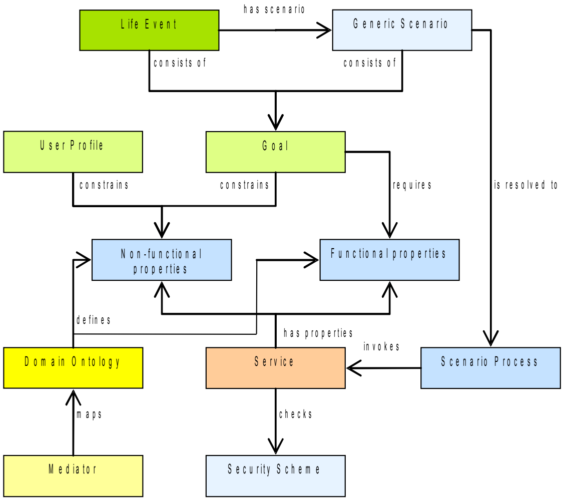


Fig 2: Logical data view on Access-eGov

provided by services. Types used to specify functional properties are defined in the domain specific resource ontologies.

Non-functional properties describe semi-structured information intended for requesters for service discovery, e.g. service name, description, information about the service provider and properties which incorporate further requirements for service capability (e.g. traditional office hours and office location, quality-of-service, security, trust, etc.). Structured non-functional properties are specified by domain specific ontologies.

The process model guides users to achieve specific goals and coordinates activities of users (citizens or businesses), traditional public administration services and web services. The process model is represented as a collection of activities designed to produce a specific output for a particular requester, based on a specific input. In that sense, activity is a function or a task that is carried out over time and has recognizable results.

### C. *Access-eGov Modules*

Most of the Access-eGov modules are derived from the needs implied by the use cases and they may be divided into three main groups: information provider related modules, information consumer related modules, and system management related modules.

#### 1) *Information Provider Related Modules*

Among the information provider related modules there are three separated modules: service annotation module, service discovery module, and service composition module.

The Annotation service module consists of a web-based application not being an integral part of the Access-eGov Infrastructure. Its main purpose is to enable domain experts to semantically describe their electronic or traditional services, by using relevant public service ontology. The web application provides also capabilities to allow annotation of traditional websites as well as an easy inspection of the existing content, which concurrently can be annotated.

Via web service interfaces, the Access-eGov Annotation module is able to access Ontology and Service Repositories within the Persistence Layer in order to register services and publish their descriptions. The creation, modification and editing of these semantic descriptions is controlled by the security subsystem.

Access-eGov system requires from the services to be semantically described and registered. For semantic description the Annotation service module uses the concepts and relationships from an appropriate ontology and to mark-up important aspects of the service or website.

Additionally that module is responsible for creating and managing goals and life events which are key idea of Access-eGov approach. Goals describe what user wants to achieve (similarly to WSMO vocabulary) and life events are situations happening to users which directly cause the need of executing some public services. Moreover goals and life events are workflow-like constructs that could be considered as outputs or interfaces provided by AeG system for users.

Service discovery module semantically matches functional properties of goals and services in the process of service discovery, in order to select services which are able to achieve the goals. Non-functional properties specified by the requester are then used to additionally filter or reorder the discovered services according to the requester preferences.

Service discovery task can be divided into two cases: “full-text search” and “semantic search”. The first case is a functionality provided in form of an interface to already existing full-text search engines in order to retrieve services and life events or goals from simply comparing the set of annotated properties. The second case is a functionality which allows to go beyond mere full-text matching for a semantic on-the-fly computation of an appropriate chain of services. The user input is then used to check preconditions and effects of registered services without invoking an already existing workflow.

Service composition module is used in case the service discovery module cannot find an atomic service which is able to achieve required goal. This module tries to orchestrate existing services to the new scenario to solve this goal. Although there are many initiatives to define industry standard languages for web service orchestration like BPEL, they have restricted capability to support only static service composition. Access-eGov Composition module provides support for dynamic composition of the services, which is not based on the static workflow pre-defined for the life event.

Automatic composition of services, which will solve these problems, is the subject of the current and future research. For this reason, current specification of the Access-eGov Composition component includes a semi-automatic approach based on the generic scenarios defined for the life event categories.

#### 2) *Information Consumer Related Modules*

Among the  information consumer related modules there are two modules: scenario execution module and personal assistant module.

Scenario execution module is responsible for executing user's scenario which is done via Personal Assistant. Service execution is invoked in case the user wants to achieve the goal, and in order to do it lets Personal Assistant start the execution of the retrieved service or workflow. The main activities of the scenario execution are as follows: invoke web service, invoke traditional service (special type of invocation, where activity is led back from the user - to the Personal Assistant. Executing of the scenario waits then for user to input the output of the traditional service.), resolve subgoal, and check timeout.

The personal assistant module is responsible for interaction with the user of the Access-eGov platform, but the user does not work directly with the module but with the Personal Assistant Client which is a web application which communicates to the Access-eGov platform and invokes its services through the module. Significant issue in this context is user's authentication through the user name and password which supports customisation of e-government services according to citizen's profile which may contain sensitive data.

#### *3) System Management Related Modules*

System management related modules cope with the core functionality of the Access-eGov platform and the only one module belongs to this category - system core module which is responsible for the interaction with the user of the Access-eGov platform and tasks relevant to the core platform functionality. That means manipulation with ontologies, connections management and security issues.

### *D. Access-eGov Components*

Each Access-eGov module consists of several components which perform its specific task and provide necessary functionality of particular parts of the modules. For example there is a group of components responsible for service annotation itself, for maintenance of life events and goals, for managing ontologies, for selecting services according to particular properties, or they provide e.g. interface to the full-text search of services.

### *E. Conceptual Architecture*

The basic functionality of Access-eGov is annotating services and store them in efficient way. Those services have to be retrieved according to certain citizen requirements and the administrators in public authorities ought to have a possibility to string such annotated services together to form new "meta services" - user scenarios. Therefore two different views may be considered: point of view of an administrator in a public institution as well as citizen's perspective.

An information provider has three main tasks, namely registering new services, annotating services and building generic workflows out of already defined services. A service consumer has two main possibilities of interaction with the platform: goal specification and request for executing the retrieved services. Communication with the platform always occurs through the personal assistant (or an equivalent user client interface).

## IV. Technologies Used

There is a variety of Semantic Technologies used for ontology creation. And then ontologies combined with semantic description are used for defining semantic web services. Access-eGov's approach required services platform to achieve semantic interoperability on different administration levels. Therefore, semantic technologies are mainly used to overcome language barriers in service description and annotation terms. In order to fulfil the goal, Access-eGov applies Semantic Web Services formalisms to create ontologies and semantically describe services. Four different formalisms were considered, namely: Web Ontology Language for Services (OWL-S), Web Service Modeling Ontology (WSMO), Web Service Semantics (WSDL-S), and Business Process Executions Language for Web Services (BPEL4WS). Before making the decision their advantages and disadvantages were considered.

### *A. OWL-S*

The structure of the OWL-S consists of a service profile for service discovering, a process model which supports composition of services, and a service grounding, which associates profile and process concepts with the underlying service interfaces. Moreover OWL-S distinguishes between atomic, simple, and composite processes. Atomic processes can be invoked, have no sub-processes, and are executed in a single step from the requester's point of view.

Two main OWL-S disadvantages are: usage of single modelling element (Service Profile) for requester and provider, and the problem with rule languages, which in spite of being recommended in the combination may lead to undecidability or leave semantics open (SWRL, DRS). There are also problems with ways of interaction between OWL and rule languages (e.g. KIF). The consortium saw a little apprehension before using OWL-S due to the fact the language must have been extended for traditional services.

### *B. WSDL-S*

WSDL-S is a small set of proposed extensions to Web Service Description Language (WSDL) by which semantic annotations may be associated with WSDL elements. WSDL-S defines URI reference mechanisms to the interface, operation and message constructs to point to the semantic annotations defined in the externalized domain models. WSDL-S defines following extensibility elements: *modelReference*, *precondition*, *effect*, *category*, and attribute – *schemaMapping*.

Significant advantage of WSDL-S is its independence on the semantic model languages. However, it has several weaknesses: new, being still in research phase approach and lacking explicit support of orchestration and choreography.

### *C. BPEL4WS*

BPEL4WS is a specification that models the behaviour of Web Services in a business process interaction. It is based on the XML grammar which describes the control logic required to coordinate Web Services participating in a process flow. An orchestration engine can interpret this grammar so it can coordinate activities in the process. BPEL4WS is a layer on the top of WSDL, which defines the specific operations and BPEL4WS defines how the operations can be sequenced.

Although BPEL4WS is very suitable for representing workflow, it needs to employ semantic into WSDL, what is a new approach. Also, there might have been problem with dealing with implementation of semantics.

### D. *WSMO*

The Web Service Modeling Ontology (WSMO) is a conceptual model for describing semantic Web Services. WSMO consists of four major components: ontologies, goals, Web Services and mediators.

Ontologies provide the formal semantics to the information used by all other components. WSMO specifies the following constituents as part of the description of ontology: concepts, relations, functions, axioms, and instances of concepts and relations, as well as non-functional properties, imported ontologies, and used mediators. The latter allows the interconnection of different ontologies by using mediators that solve terminology mismatches.

Goals specify objectives that a client might have when consulting a Web Service, i.e. functionalities that a Web Service should provide from the user's perspective. Goals are characterized by a set of non-functional properties, imported ontologies, used mediators, the requested capability and the requested interface.

A Web Service description in WSMO consists of five subcomponents: non-functional properties, imported ontologies, used mediators, a capability and interfaces.

A choreography description consists of the states represented by ontology, and the if-then rules that specify (guarded) transitions between states. The ontology that represents the states provides the vocabulary of the transition rules ant contains the set of instances that change their values from one state to the other. Concepts of an ontology used for representing a state may have specified the grounding mechanism which binds service description to the concrete message specification (e.g. WSDL). Like for the choreography, an orchestration description consists of the sates and guarded transitions. In extension to the choreography, in an orchestration can also appear transition rules that have the invocation of a mediator as post-condition. The mediator links the orchestration with the choreography of a required Web Service.

Mediators describe elements that aim to overcome structural, semantic or conceptual mismatches that appear between the different components building up a WSMO description.

After analysis WSMO proved to be formalism having the biggest range of advantages, as relying on loose coupling with strong mediation, combining conceptual modelling and rules, and providing opportunity for sophisticated goal-oriented discovery.

One of WSMO advantages crucial for Access-eGov is very clear distinction between functional and non-functional properties which are used within Access-eGov project very widely. WSMO enumerates all the relevant functional properties (for example Web service has Capability and Interface as its functional properties) and it allows flexible and easy to extend sets of NFPs everywhere. In WSMO conceptual model it is possible to enumerate all the parameters that are functional properties for the aim of WSMO. It is also possible to add different NFPs according to the own needs of the user.

WSMO provides a consistent conceptual model for the semantic description of web services, with the inclusion of mediators and the distinction between goals and services. In addition, the WSMO conceptual model fits best the proposed architecture and functionality of Access-eGov system.

However, some parts of the specification haven't been finished yet, what was considered as disadvantage, but consequently it offers possibility of specification development and conducting research. Moreover, orchestration and choreography is based on the abstract state machine where workflow is encoded and it also must be extended for traditional services.

Despite these inconveniences, consortium has decided to use WSMO technology as best suitable for Access-eGov's approach.

## V. Access-eGov Ontologies

### A. *Conceptual View of Ontologies*

Access-eGov ontologies are utilized to semantically express real-world concepts in a way defined and agreed upon by communities of users. “Ontology” in technical terms constitutes a formal specification of a shared conceptualization. Ontologies define an agreed common terminology by providing concepts and relationships between these concepts. In order to capture semantic properties of relations and concepts, ontology also provides a set of axioms (i.e. logical expressions in some structured language). Access-eGov uses three basic ontologies: life events ontology, service profiles ontology, and Acces-eGov domain ontology. Structure of ontologies is illustrated in the figure 3.

The domain ontologies are considered lower level ontologies within the system. They describe all the relevant pieces of domain information related to user’s scenarios. That means they describe functional and non-functional properties of a particular service. They are web based ontologies that are not necessarily relevant to the web services. The domain ontology describes  the lower (i.e. technical) level of the Access-eGov system.

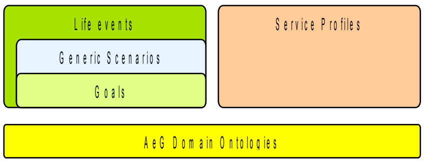


Fig 3: Conceptual data view in Access-eGov

The domain ontologies contain conceptual descriptions of domain-specific information for the pilot applications. It includes the concepts describing various forms, documents, certificates, location constraints, fees, questions, notification messages, etc., that are necessary to model the inputs and outputs of the provided governmental services.

The other two ontologies are “Life events” and “Service Profiles” ontologies which describe more abstract data. They are not simple web ontologies, but extended with semantic descriptions of possible life events (Life Events ontology) or (web) services (Services Profiles ontology). They denote more abstract system levels just as service description.

The Life events ontologies contain conceptual descriptions of life events, complex goals (also referenced as generic scenarios), and elementary goals for the pilot applications. The elements of the ontology are expressed by the WSMO choreography and orchestration interfaces.

The life events and goals described in the Life events ontology are used in the Personal Assistant client tool. The life events and goals of all the Access-eGov pilot applications specify a process model that will be composed and executed by the inner components of the Personal Assistant client according to the interactions with users.

All three Access-eGov ontologies describe several aspects and levels of the same real world data. All of them denote services (electronic or traditional) and the way of their usage.

### B. *Formalisms Used*

The WSMO conceptual model was adapted and modified to meet the requirements of the life event approach to modelling governmental applications. Thus the resource ontologies were formalised and implemented using the WSML (Web Service Modelling Language) representation. More precisely toolchain for ontology manipulation was designed, consisting of the specialised Annotation Tool and of the third-party WSMO Studio environment. The Annotation Tool was developed as a web application for user-friendly semantic annotation of governmental services. This tool, together with the resource ontologies, was tested by all the public administrations involved in the Access-eGov project.

The conceptual model contains a set of relevant entities - concepts, relations, properties, constraints, etc., that can serve as building blocks for the implementation of the system components as well as for the semantic annotation, i.e. the formal representation of potentially very complex governmental services and their relationships.

The WSMO conceptual model provides following top-level elements: ontologies which provide terminology used by other elements, web services which represent computational entities able to provide access to services, goals which describe aspects related to user's requirements, and mediators which describe elements handle semantic interoperability problems between WSMO elements.

Structural relations between the elements in the proposed conceptual model are depicted in figure 4. The parts reused from the original WSMO model are marked with grey colour.

### C. *Design of Ontologies*

The ontology-like structure needed to be formalised and expressed by WSML statements. It required fixing the meaning of the terms and relations defined in the controlled vocabulary, as well as verifying that the formal meaning reflects the informal description in the glossary.

The concepts and their relations were modelled by the following expressions:

```
concept ConceptName
  relationName RelaedConcept
```

For example, a hierarchy of certificates can be expressed in WSML notation as follows:

```
concept Certificate
  subConceptOf Document
concept Birth_certificate
  subConceptOf Certificate
concept Marriage_certificate
  subConceptOf Certificate
```

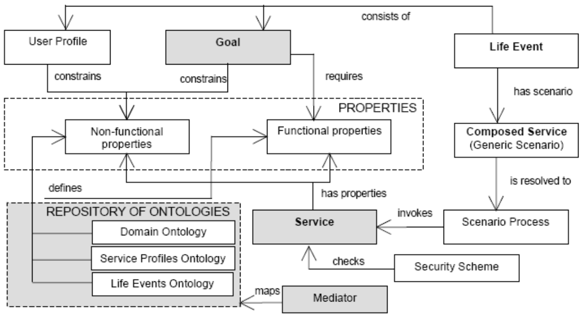


Fig 4: WSMO conceptual model adapted for the Access-eGov system

In addition, the external ontology resources, identified as relevant for the given domain, were used to standardise the ontology structure and to achieve a consistency between the semantic descriptions. The attributes of concepts were modelled as NFPs. Since the attributes are displayed in the client-side tools, they need to be localised to the proper languages. The localised values are modelled by Dublin Core's dc#title statements.

## VI. Access-eGov most important facts

Access-eGov provides a specific and new approach to the administrative issues – namely user's point of view. Realisation of such a perspective uses user's goals and life events, which exactly describe user's needs and take into consideration specific user's conditions deriving from user's characteristic context.

Executive scenario is dynamically composed on the basis of user's requirements as well as service capabilities. That is first of all user-friendly, and does not use administration-centric point of view which is not understandable by common users. Moreover, dynamic composition helps to create particular user's path consisted of separated atomic services, which are to be executed via Access-eGov platform in the meaning of web service execution or waiting for user input after realising traditional services has been completed.

Such an approach is directed towards the users – citizens or businesses, to facilitate their interactions with public administration and to make government services interoperable.

Administration-centric approach is comfortable only for institutions, because considers realisation of particular services, and user-centric approach constructs whole, unified processes built from atomic services. Such a process meets the whole path of realisation of particular life event, complying also tasks not related to public administration, but essential to complete user's goal.

The second important Access-eGov value added is usage of WSMO technologies, which are less popular among EU projects. It turns out that WSMO approach of user goals and service capabilities is strictly tailor-made for user-centric approach.

## Acknowledgment

The author expresses thanks to Access-eGov consortium for being involved in the Project and consequently having possibility of writing this article using Access-eGov deliverables as sources of knowledge.

## References

[1] Access-eGov, "State-of-the-art Report" D2.1, May 2006.

[2] Access-eGov, "Access-eGov Platform Architecture" D3.1, February 2007.

[3] Access-eGov, "Access-eGov Components Functional Descriptions" D3.2, March 2007.

[4] Access-eGov, "Public administration resource ontologies" D7.1, November 2007.

[5] Peristeras, V., Tarabanis, K.: Reengineering the public administration modus operandi through the use of reference domain models and Semantic Web Service technologies. In: *Proceedings of the 2006 AAAI Spring Symposium on The Semantic Web meets eGovernment* (Stanford University, March 27-29, 2006), Technical Report SS-06-06, AAAI Press, Menlo Park, CA, 2006,

[6] Roman, D. et al: D2v1.0. Web Service Modeling Ontology (WSMO). WSMO Working Draft, 20 September 2004. Accessible at http://www.wsmo.org/2004/d2/v1.0/ [Last accessed in October 2007].

[7] Wang, X., Vitvar, T., Peristeras, V., Mocan, A., Goudos S., Tarabanis, K.: WSMO-PA: Formal Specification of Public Administration Service Model on Semantic Web Service Ontology, Hawaii International Conference on System Sciences (HICSS), Waikoloa, Big Island, Hawaii, 2007.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# In Search of Values in Internet Shopping–Method and Preliminary Research

Jacek Wachowicz
Gdańsk University of Technology,
Email:
jacek.wachowicz@zie.pg.gda.pl

Piotr Drygas
Poznań University of Economics,
Email:
Piotr.Drygas@ae.poznan.pl

***Abstract*—Internet shopping becomes more and more popular. One of the most important questions for internet enerpreneurs seems to be how to encourage users to spend money. This is strictly connected with users' motivations. Therefore arises a strong need of learning, what they are and which factors, represented in product's features, are influencing them. One of most promising techniques in this field seems to be Hierarchical Value Maps. This paper describes the method and outcomes from preliminary research.**

## I. Introduction

As the number of internet shops is growing it is constantly harder and harder to gain customers that would like to do shopping. Many research is trying to find out what are motivations for certain human actions. One of newest methods, considered to be a promising one, are Hierarchical Value Maps. In this paper authors describe them – and present preliminary data, that may be used in the method.

## II. Value Maps

The Hierarchical Value Maps derive from the Means-end theory, which assumed that there always are some hierarchies in all human actions – so are in products and services perception and valuation. Users gain them from attribute experiences, which drives (through consequences) to personal values held by an individual. To have them illustrated better, later introduced graphical method, known as laddering, turned to be very helpful. Usually, in this technique, attributes are presented at bottom layer – and they reflect all important, perceived, features, that may influence users attitude to this product or service. Of course most important for users are their personal goals and habits. Therefore they are usually presented on top. Needless to say, in most cases they are different from product's features. But they have to depend (directly or indirectly) on product's features. These dependencies are usually shown in the middle layer – and in value maps are called consequences. The graphical form of these dependencies is known as a Laddering Technique. It became a commonly used form in search of consumers behavior structures and is widely used, especially in marketing research. Its final graphical form is known as Hierarchical Value Map (HVM). An example Hierarchical Value Map is shown on fig. 1 [2]

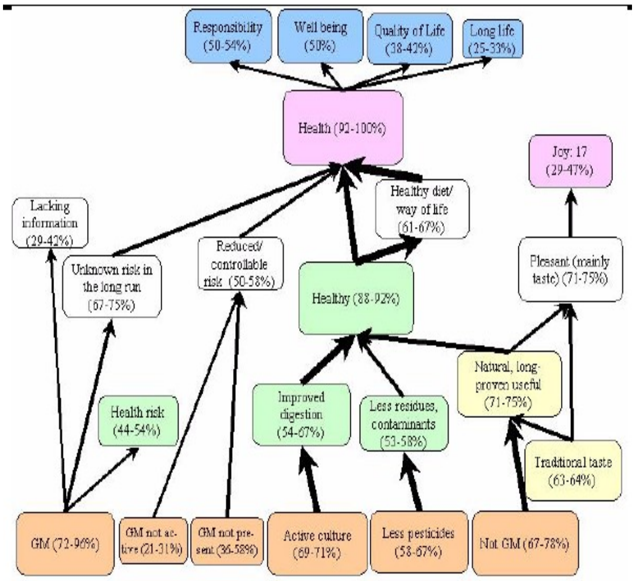

Responsibility (50-54%)
Well being (50%)
Quality of Life (38-42%)
Long life (25-33%)
Health (92-100%)
Joy: 17 (29-47%)
Healthy diet/ way of life (61-67%)
Lacking information (29-42%)
Reduced/ controllable risk (50-58%)
Unknown risk in the long run (67-75%)
Healthy (88-92%)
Pleasant (mainly taste) (71-75%)
Natural, long-proven useful (71-75%)
Health risk (44-54%)
Improved digestion (54-67%)
Less residues, contaminants (53-58%)
Traditional taste (63-64%)
GM (72-96%)
GM not active (21-31%)
GM not present (36-58%)
Active culture (69-71%)
Less pesticides (58-67%)
Not GM (67-78%)

Fig. 1: Sample Hierarchical Value Map

The Laddering Technique process that leads to such diagram, usually begins with semi-standardized, in-depth review series. During review session researcher should ask questions that may discover attribute-consequence-value chains. One of important difficulties is 'transparent' reviewing, which means that researcher can not suggest answers. This requires special training in question formulation, as commonly people tend to ask questions in such a way, that simple confirming is (awaited) positive information. Moreover, researcher needs to drive the review so that reviewed person may take initiative (which helps to find unexpected factors and value chains). It is very import ant, that each time an answer should be a starting point for researcher in search for new attribute-consequence-value chains. Therefore a very important is asking questions in such a way, that all vital associations may be discovered as a part of individual reflection of a person being under research. Achieving that requires all reviews to be individual and conducted by skilled researcher.

Obtained answers are presented as graphical chains between attributes, consequences and values later on. Connections are graphically represented by arrows, showing rela-

978-83-60810-14-9/08/$25.00 © 2008 IEEE

tions between linked elements. Usually, attributes come from features of product or service and may be directly observed and measured. Normally, they don't affect users values or preferences in a direct way. But theory assumes, that the product (or service) perceived usability reflects level of accomplishment of user's preferences or values (as users perceive products by satisfying user's needs) – and they come from features that are satisfied by product's attributes.

Consequences create a middle layer – between attributes and values. But what seems to be most important for user is a way in which physical attributes satisfy his/hers needs and respond to user's values (and not physical attributes of products). Therefore physical attributes need to be considered to be secondary in a process of product's definition formulation – and defined as implications of user needs reflected by consequences. Later on, consequences may be additionally described by weights reflecting importance of each consequence in order to fill user's value. The whole net of consequences allow to track which attributes are important in user's needs satisfying.

## III. Research

The above described technique is helpful in finding what is important for internet buyers. It is good to have a general outline of internet buyers' positions for having an idea how to conduct individual reviews. Therefore authors have done initial internet review, which outcomes are presented in this paper. They shall be used as a starting point in future, for individual reviews.

The research was done through a dedicated Web-questionnaire. Taking part in this research didn't require entering any personal data, unless person under research wanted to be informed of the current research statistics.

The questionnaire was split into two parts: main and demographic data imprint. 438 answers was collected during this research, of which 437 was accepted for further analysis (after verifying collected answers). On charts 2 to 4 characteristics of colleted group is presented (according to imprint).

The population and characteristics of group under research was not formally constructed to be a general one. However, it is satisfactory coherent with other researches and therefore one can assume its correctness. It was decided, that its outcomes can be used for further conclusions.

The research problem was set to find what factors (or instruments) are important during the process of making a decision on buying in internet shops. The starting point for this research was finding, that 87% of respondents has ever made a purchase in an internet shop. The frequency of internet purchases distribution is shown on fig. 5

Mostly respondents did their Internet shopping in two to four shops (50%) and next quarter of them – in less then 10 shops (26,6%). Only 9,5% of respondents did their shopping in more ten 10 shops.

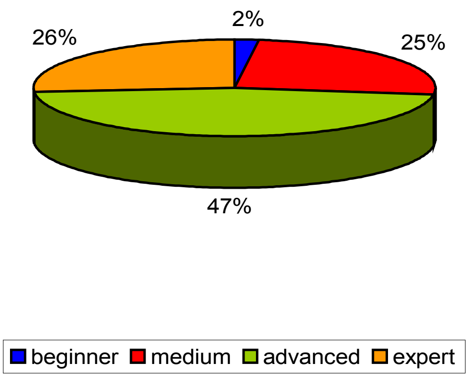


Fig. 2: Age structure of respondents

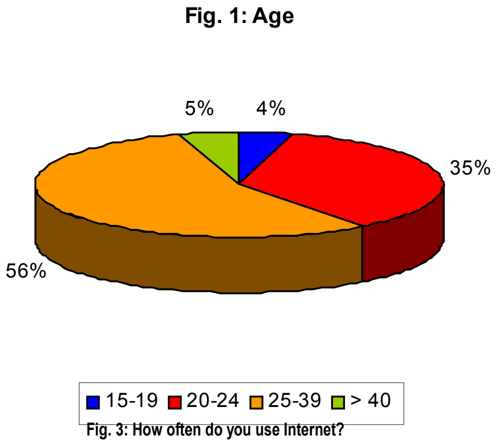


Fig. 3: IT competences of respondents

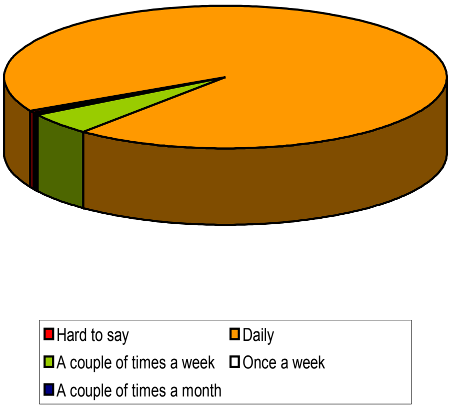


Fig. 4: How often do you use Internet?

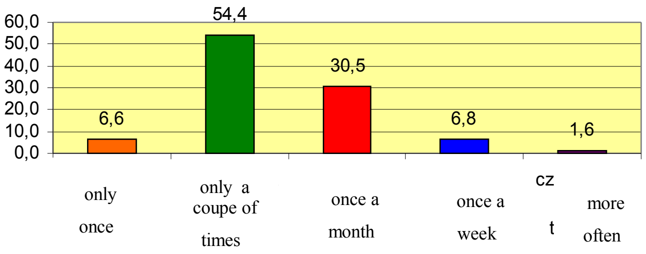


Fig. 5: How often do you buy products/services via Internet?

Next, respondents were asked about important features of internet shops. The most important one turned to be its functionality (27,2% of answers). Other important factors are company's offer (18,7%), delivery options (11,2%), price policy (10,6%) or communication with company (10,1%). The loyalty programs gained only 3,9% of votes. This shows that consumer loyalty may be strengthened with other then implicit instruments (like points, rebates, rewards).

It is worth to remind in this point, that internauts tend to build their trust with communication means, including personalization. Of course every internet shop would like customers to get back. So the next question was what is driving clients towards returning to the virtual shops. An answer to this question would be at a time a verification of basic features important for buyers. This verification was done in research a question: 'Indicate factors that may positively influence your decision on next transaction within the web site'. Outcomes are presented in Tab. 1

TABLE 1.
FACTORS INFLUENCING REPETIVE BUYING IN AN INTERNET SHOP.

| Factor | % |
|---|---|
| Price | 16,42 |
| Duration of Transaction | 12,56 |
| Delivery | 9,88 |
| Consumer service | 7,87 |
| Satisfactory previous transaction | 7,71 |
| Detailed description of the product | 5,86 |
| Oferring | 5,36 |
| Discounts for long-standing customers | 5,36 |
| Functionality | 4,52 |
| Communication | 4,52 |
| Quality of goods | 4,36 |
| Credibility | 3,52 |
| Promotions | 2,51 |
| Loyalty programs | 1,68 |
| Bonuses for customers | 1,34 |
| Diversity of payment methods | 1,34 |
| Seller 's competences | 1,17 |

As it might be seen in Tab. 1, factors that influence customers' return decision mostly are price, transaction time, conditions and delivery means, customer service and previous experiences. The fact, that four out of five of them were previously pointed as possible means of repetitive shopping, confirms correctness of presented conclusions. Moreover, we can consider correct consumer service to be a sine qua non condition of any consumer relations.

Considering that all these problems reflect in communication, it should be of great importance to find consumers' opinion on the most important issues of communication. That reflects in answers to the following questions:

1. Do you see need for contact with a seller before buying in an e-shop?
2. Do you expect seller to contact with you after buying in an e-shop?
3. Do you think that seller should investigate if selling process (goods, delivery, payment etc.) was satisfactory?

The outcomes are presented on Fig. 6.

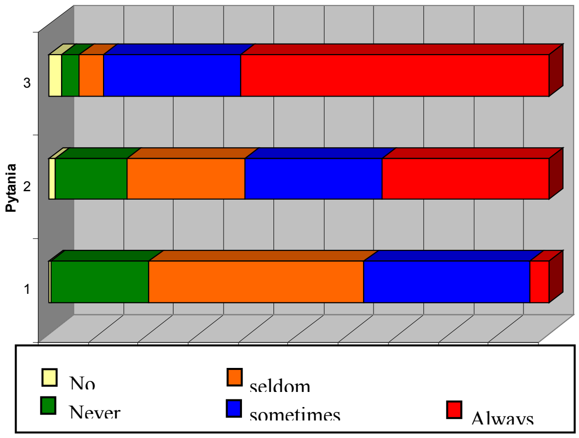


Fig. 6: Communication importance In customers' opinions

The outcomes show large consumer communication needs. Probably this is strictly connected with Internet specifics, which is highly of communicational and informational nature. The research shows dependency, which is very often overseen – both in literature and in practice – that communication importance increases as selling process advances. Before buying sees communication rarely (43,03%) or never (19,57%). Only 3,93% of buyers tends to find any form of contact before buying. Analyzing answers to the question about expected sellers contact after sale one can observe a strong shift in consumer attitude. In this case most of internauts (33,48% in every case and sometimes – 27,34%) do expect some form of contact – and therefore importance of communication increases. Moreover next 23,48% of internauts do expect some contact only rarely after sale.

## PRELIMINARY MODEL

After preliminary research it may be constructed preliminary Hierarchcal Value Map presenting consequences (relations) between universal web-product's features and abstract buyers values. Its first iteration of HVM is presented on fig. 7.

From this model we can see, that three features – quality, delivery and service seem to be most influential. There is also one identified feature – protection – which was not directly pointed by respondents, but is necessary for building expected by users value, safety namely.

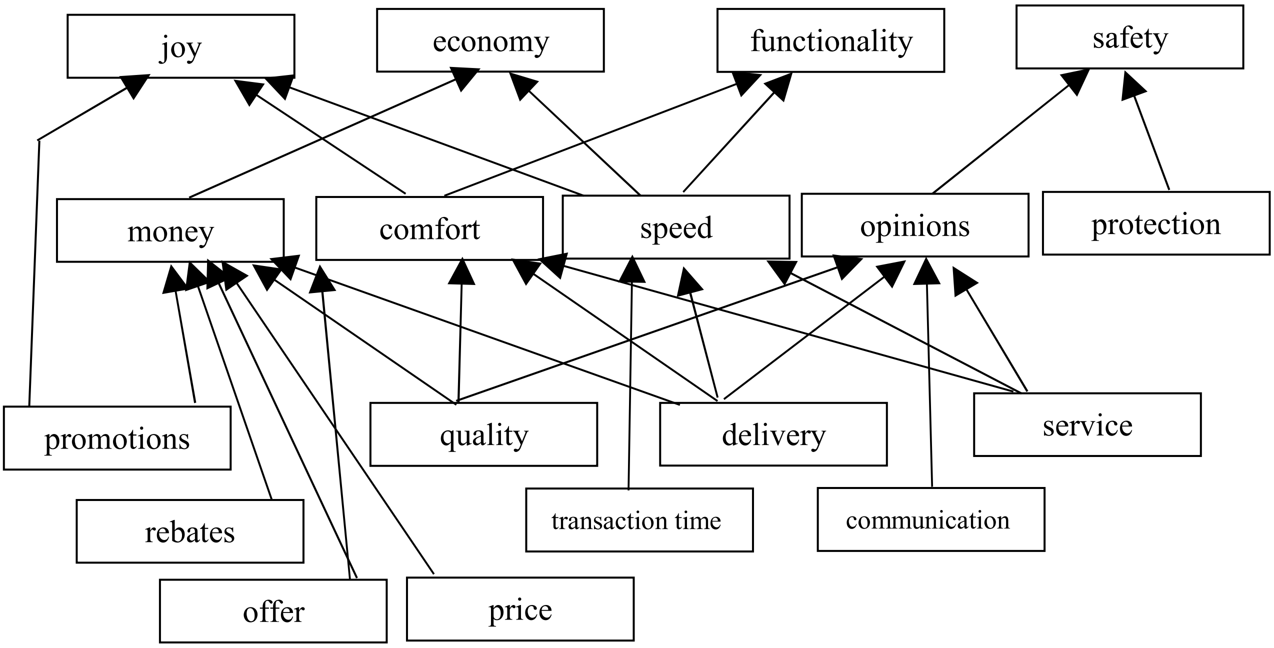


Fig. 7: Preliminary Hierarchical Value Map for web-enabled product or service

Of course this should be verified with a number of individual reviews – and for having them more understandable for people being reviewed – it seems it should be driven for a couple of characteristic web-enabled products and services.

## References

[1] Banister D, Mair J. M. M., The evaluation of personal constructs. London Academy Press, 1968

[2] Boecker A., Perceived Risk is Important for Consumers' Acceptance of Genetically Modified Foods, but Trust in Industry not Really: A Means-End Analysis of German Consumers, 99th EAAE Seminar 'Trust and Risk in Business Networks', Bonn, Germany, February 8-10, 2006, s. 7, http://ageconsearch.umn.edu/bitstream/123456789/29171/1/sp06bo03.pdf

[3] Gengler C. E, Reynolds T. J., Consumer understanding and advertising strategy: Analysis and strategic translation of laddering data. Journal of Advertising Research, 35 (july/august) 1995, pp. 19-33

[4] Reynolds T. J., Gutman J., Laddering theory, method, analysis and interpretation. Journal of Advertising Research, 28 (February/march) 1988, pp. 11-31

[5] Reynolds T. J., Dethloff C., Westberg S. J., Advancements in laddering. In: Reynolds T. J., Olson J.C. (Eds.) Understanding Consumer Decision Making–The Means-End Approach to Marketing and Advertising Strategy. Lawrence Erlbaum Associates, Mahwah, NJ 2001, pp. 91-118

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 433 – 436

ISBN 978-83-60810-14-9
ISSN 1896-7094

# V-Commerce: The Potential of Virtual Worlds in Services

Urszula Świerczyńska-Kaczor
University Jana Kochanowskiego
in Kielce
ul. Żeromskiego 5, 25-369 Kielce,
Poland
Email: swierczynska@pu.kielce.pl

*Abstract*—**The author assesses the suitability of virtual worlds for building effective selling environment . In analysis key features of virtual worlds are discussed and compared with the way of functioning the other type of virtual communities—social networks web-site.**

## I. Introduction

WHEN the idea of embedding virtual communities in business appeared in the mid 90's [1] the numerous virtual communities have grown in Internet space. N owadays many companies apply virtual communities [1] into their functioning, mostly in promotion and R&D, often resorting to social networks or sponsored web site. Among many different types of virtual communities, virtual worlds based on the new technology previously created for entertainment start playing important role in many business areas including distribution. Due to better description of virtual words even new marketing terms were invented like 'v-commerce' or 'v-CRM' [17]. Because 'pure' services [2] can be sold without material component, virtual worlds offering direct real-time communication with 3D graphics, become perfect venues in the Internet space for meeting customers and companies. Virtual world such as Second Life is still emerging new technology and their potential business users, such as companies operating in a real market, have limited knowledge about implementing virtual worlds' into their business environment. Many entrepreneurs perceive virtual worlds only as 'games', not discerning their potential in business areas. This article is focused on analysis of the role of virtual worlds in services. The analysis, based on qualitative research - netnographic studies, highlights the differences between implementation 'traditional' social networks and virtual worlds as a distribution channel.

## II. The main features of virtual worlds—example of Second Life (SL)

Second Life (SL) is a well-known virtual world with about 1.2 million of users, mostly from United States (about 36%) and from Germany, France, Brazil, Japan and United Kingdom[3]. Second Life is an example of Massively Multiplayer Online Games (MMOGs)[4], but significantly different from the goal-focused on-line games such as racing or fighting. This virtual world is also called 'metaverse' [7] in order to distinguish Second Life from non-social games. Second Life reflects 'real' life, therefore the avatars spend their money on entertainment such as shopping, going to clubs, disco, museums, parks.

Virtual world is an interface for people who want to meet, talk on-line with other users and play by creating their own world. SL also works as an Internet tool for companies looking for their potential customers, although the customers in Second Life (and virtual worlds as general) exist as avatars.

Depending on the connection with real world, the companies operating in Second Life can be classified into two groups:

1) Companies existing in the real market which aimed to strengthen their brands using virtual worlds:

- manufactures eg. Fiat, Peugeot, IBM, Adidas[5], Sony, Reebok, Sun Microsystems
- retailers e.g. Circuit City, Sears,
- service companies e.g. AOL, hospital: Palomar Pomerado Health [5]
- media ex. Reuters, National Public Radio NPR, Polish Radio Bis, Warner Bros

2) Companies which operate only in the virtual market e.g. language schools providing lessons via chat or voice, companies creating digital houses, garments etc. for avatars.

The varieties of business existing in Second Life is incredibly diverse: from companies connected with creation in SL such as custom avatar designer, jewelery maker, architect, freelance scripter, theme park developer [8] to companies using interface as a platform for delivering professional services such as education or architecture.

The business in Second Life flourishes due to the possibility of trading currency which is used inside the game (Linden

[1] The terms 'virtual community', 'virtual society' or 'social networks' are still unclear and authors use different criteria in analyzing this phenomenon

[2] Such as educational services or lawyer's advice

[3] Poland is 14-th in the ranking of active accounts with 4.426 avatars – less than 1% of all residents – April 2008 [9]

[4] There.com or Entropia Universe are other examples

[5] Adidas promoted their running shoes – the a3 Microride – creating the experimental store in Second Life, the project was successful: three out of every five visitors bought the product, virtual sales alone earned back 14.5% of total investment, also 3.750 on-line articles and TV interviews plus extensive magazine coverage were generated [15]

978-83-60810-14-9/08/$25.00 © 2008 IEEE

Dollars). It means that an entrepreneur who gains profit selling products or providing services inside a game, can exchange the earnings (Linden Dollars) into real money – American Dollars.

In the table 1 (see below) there are some main descriptions of Second Life and examples of social networks: Facebook, MySpace (global scale) and fotka.pl, Moja Generacja.pl, nasza-klasa.pl (regional market).

The products or services sold inside virtual communities can be classified into two categories [see the fig. 1]:

1. Products exclusively for entertainment. In Second Life these are new garments, houses, boats, cars and other products for avatars. In Facebook or in nasza-klasa there are some digital present for friends (like small pictures). In fotka.pl the user can buy 'being a star'. These products are useless for members of the community outside the game.

2. Products that can be used out with the gaming environment. In this case the virtual world is only an interface (environment) in providing services or selling products - these products/services could be sold in similar way using alternative Internet channels such as the web site. Providing language lessons or educational courses are the examples of services, but also digital products such as e-book, music, photos can be sold in this way.

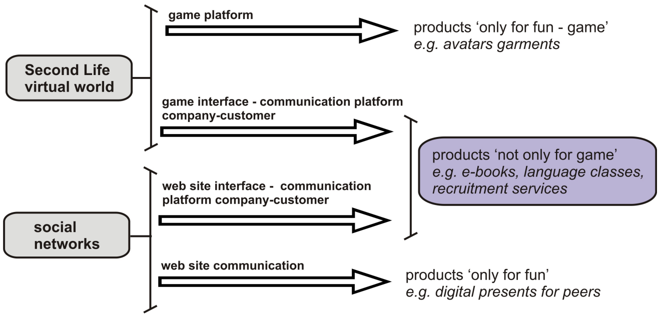

game platform
products 'only for fun - game'
e.g. avatars garments
Second Life virtual world
game interface - communication platform company-customer
products 'not only for game'
e.g. e-books, language classes, recruitment services
web site interface - communication platform company-customer
social networks
web site communication
products 'only for fun'
e.g. digital presents for peers

Fig 1. The link between Second Life and social networks as distribution channels

Although Second Life is still in the early development phase, there are some examples of traditional companies using SL in their distribution. They successfully implemented SL in educational courses, consulting services, engineering and architecture projects or recruiting staff (see examples below)

1. Many schools use Second Life in teaching:
   (a) 43 universities in the United Kingdom work with Second Life organizing different kind of activities [6]
   (b) Insead provides MBA lecturers [20],
   (c) Ball State University in Indiana [19] , Montana State University-Bozeman and Harvard Law School [10] use the SL as a e-learning platform,
   (d) Standford University and University of Houston examine different aspect of functioning virtual world [10]
   (e) A group of schools in Barnsley resort to a virtual word to teach children writing and comprehension skills [21]
2. Non-profit institutions such as museums, art galleries, public radio organize educational events e.g.:
   (a) NPR Radio organizes Science Friday
   (b) The Royal Liverpool Philharmonic gives performances [4]
3. SL is useful in HR projects:
   (a) Novartis and Johnson&Johnson [4], Cisco Systems, BMW and Vodafone [23] use SL in the program of training their staff and company
   (b) Wipro Technologies receives submitted resume at Wipro's virtual campus [2]
   (c) Dutch bank ABN Amro is using the virtual world for one-to-one meetings with prospective employees [18],
4. Some engineers resort to Second Life as a platform for projects: 3D graphic is more convenient in presentation for clients than traditional architectural programs (e.g. Second Life allows customers experience walking through the house) [22]

Among many variables determining the potential of virtual communities in distribution, two groups of factors are especially important (see fig. 2):

1. factors connected with the technological environment
2. factors connected with characteristics of users

Apart from the characteristics of the community such as the age of members, the number of users or the intensity of communication, also the technical infrastructure affects the process of delivering services. Using only chat or instant

Table I.
Comparison of virtual worlds and chosen social networks

| Feature | Second Life | Facebook.com | Fotka.pl | Moja Generacja.pl | Nasza klasa.pl |
|---|---|---|---|---|---|
| Type of community | Virtual world | Social network - global | Social network - regional | Social network - regional | Social network - regional |
| Time founded | 2003 | 2004 | 2001 | 2006 | 2006 |
| Number of members [9], [11]-[13] | 13,7 mln users; 1,2 mln users logged during the last 60 days | 70 mln us | 3 mln users | 6 mln users | 6 mln users |
| Availability of Polish language | No | Yes, since December 2007 | Only Polish | Only Polish | Only Polish |

messaging is a more limited way of communication between entrepreneur and potential customer than voice communication.

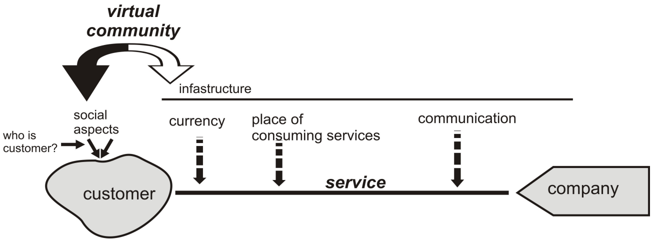


Fig 2. The factors influencing the process of delivering services

The following differences between social networks web site and virtual words can be pointed out [see table 2]:

1. Interaction with customers
2. Potential of creation
3. Currency

## III. Interaction with customers

The most significant advantage of Second Life is expressed by words of vice president ABN Amro's Bank: "The 2D Internet is excellent for simple human-machine interaction, but the 3D social Internet enables ready human-to-human interaction, or anonymous avatar-to-avatar communication" [3]. In the virtual word the environment allows the Internet user to see others' gestures, express emotions, hear voice and give indication that they really interact with others. Moreover the nature of interaction between avatars reflects the nature of human being interaction such as social norms of gender (the interaction between male and female avatars), interpersonal distance and eye contact [24]. On traditional 2D web-site of social networks, even if the communication goes smoothly, there is a significant delay in interaction.

The next issue is users' privacy. On most social networks web sites the posts and discussion are visible for all members. In virtual worlds the communication can be 'one-to-one' or provided among a chosen group.

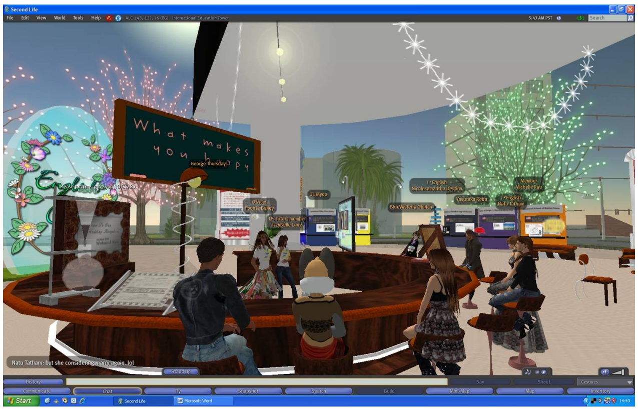


Fig 3. The English language course for avatars in SL – example of delivering service inside virtual world

## IV. Potential of creation – building effective selling environment

The unlimited potential of virtual worlds in creating digital space leads to the situation when the customers-avatars can be 'immersed' in an environment which is far different from traditional distribution channels. The digital world allows entrepreneurs to create the venue for meeting with their customers which is the most preferable for selling a particular brand and does not reflect a traditional office. The employees can meet customers in a space craft or old castle – depending which architectural surroundings better express the nature of the company. The company can enhance the level of interaction using music, videos, podcasts, interactive presentation and voice communication. There is also no technological barrier in teleportation - avatars can easily move in a particular place. The creation into virtual worlds seems to be limited only by imagination of the entrepreneurs. But the aim of this marketing is to put customer into the space which strongly enhances consumer's involvement in purchasing and their engagement into brand.

Consumption inside SL can be perceived as a substitute to consumption in the real world without causing an environmental effect. Because customers do not travel in order to receive services, companies do not build expensive headquar-

Table II.
Key features of analyzed virtual communities in distribution services

| Feature | Second Life | Facebook.com | Fotka.pl | Moja Generacja.pl | Nasza klasa.pl |
|---|---|---|---|---|---|
| Tools of communication | Chat, voice communication, messages send to avatar's e-mail, notes | E-mails and instant messaging, forums | Chat, e-mails, forums | Forums, instant messaging (Gadu-Gadu), recording audio information, Internet phone (Gadu-Gadu – is not fee of charge) | Forums, e-mail |
| Time of communication | Mostly real-time | Mostly asynchronous, real-time possible | Mostly asynchronous, real-time possible | Mostly asynchronous, real-time possible | Asynchronous |
| Currency | Not real currency, Linden Dollars | Dollar | Polish zloty | Polish zloty | Polish zloty |
| Customers appearance | Avatars 3D | Name/nick – image 2D | Name/nick – image 2D | Name/nick – image 2D | Name/nick – image 2D |

ters, the ecological footprint of providing services in the virtual world is significantly lower than in the traditional market [14] and for some customers there could be an important reason for purchase.

Also the environment created in virtual worlds is much more friendly for visually impaired people than traditional web sites, including social networks web sites.

## V. Limitation of virtual worlds

### A. Moving avatars

There is difficulty in 'moving' avatars between different virtual words – the market is limited to the members of users of a particular virtual world. In other Internet services such as instant messaging, Internet phone and social networks the process of integration has just started – allows the customers to integrate their content from different site e.g. Facebook allows to implement user's content from del.icio.us. Naszaklasa integrates contact from Gadu-Gadu etc.

### B. Currency

Using internal currency, e.g. Linden Dollars, can be perceived as a limitation of virtual worlds compare to social networks. Facebook sells its virtual gifts for American dollar (even at Polish language site), although there is a plan for introducing on-line currency at Facebook [16].

### C. Profile of users

Many social networks attract users from a particular regional market and the potential customers are people with the same cultural and sociological background. On the contrary Second Life has users from all nations, although some markets such as US, Western Europe and the Far East have major representation. Second Life is mostly suitable for companies which selling services at global scale.

## VI. Conclusion

Service companies which resort to virtual worlds receive an immersive environment, which allows using real-time, three dimensional audio and video and software to enable companies to communicate and interact with individuals or groups of customers. Due to the potential of creation in virtual worlds companies can build the selling environment which strongly enhances consumer's involvement, their perception of brand and influences the satisfaction from products. Nowadays the main limitation of using virtual worlds is the lack of connection between different platforms. Moreover in virtual worlds many business aspects still become unclear e.g. law regulation (taxes) or the consumers' rights. Social networks compare to virtual worlds are more useful as a promotion board and the selling environment is limited to digital gadgets or a few items connected with web site. Although there are a few examples of companies which resort to social networks as a distribution channel [6] the 2D Internet web-site has many disadvantages compared with 3D virtual worlds.

## References

[1] Armstrong A., Hagel J. III "The Real Value of On-Line Communities" in Business Harvard Review, May 1996

[2] Business Today, November 18 2007, "3D Marketing Wipro hopes a virtual presence will attract talent" Rahul Sachitanand

[3] Computer Weekly 11/6/2007—„Virtual worlds are 2008's 'breakthrough technology"

[4] Donahue Marylyn "Setting up shop on Second Life" in *Pharmaceutical Executive*, November 2007 Consultants Confidential Supplement, vol. 27

[5] Gaudin Sharon "Real-World Hospital Makes Virtual Debut in Second Life" in *Computerworld* 03/03/2008

[6] Information World Review - „Is Virtual a Virtue in Scholarship?"; December 2007, www.iwr.co.uk/Academic, 2007 Incisive Media

[7] http://en.wikipedia.org/wiki/MMOG#MMO_Social_game, 25.05.2008

[8] http://secondlife.com/whatis/businesses.php 25.05.08

[9] http://secondlife.com/whatis/economy_stats.php, 25.05.08

[10] http://secondlifegrid.net/how/education_and_training, 27.05.2008

[11] http://www.facebook.com/press/info.php?factsheet,

[12] http://www.fotka.pl/info/o_stronie.php,

[13] http://www.mojageneracja.pl/o_nas - 21 May 2008

[14] Lin Albert C. *Virual Consumption: A Second Life for Earth?*— Brigham Young University Law Review; 2008, vol. 2008, issue 1

[15] New Media Age, 11/15/2007, "Consumer Products and Services" Supplement

[16] New Media Age, 3/27/2008, "Facebook Calls for Online Currency"

[17] Research Technology Management - "Firms Entering Virtual Worlds" Jan/Feb 2008, vol. 51, issue 1

[18] Riley John „Virtual worlds are 2008's 'breakthrough technology" in *Computer Weekly* 11/6/2007

[19] Robins S., Deb Antoine, „Second Life w nauczaniu", Szkoły Głównej Handlowej w Warszawie, wywiad, http://www.e-mentor.edu.pl/artykul_v2.php?numer=21&id=473

[20] Sarvary Miklos „Metaświat: telewizja przyszłości?" Harvard Business Review Poland, February 2008

[21] Thomson Rebecca "Barnsley schools use virtual world to teach reading and writing skills" in *Computer Weekly* 2/26/2008

[22] Traum Matthew J. "Second Life: a Virtual Universe for Real Engineering" *Design News* 10/22/2007, vol. 62, Issue 15

[23] Weekes Sue "Get a SECOND LIFE" in *Training & Coaching Today*, Nov/Dec 2007

[24] Yee Nick, Bailenson Jeremy N., Urbanek Mark, Chang Francis, Marget Dan "The Unbearable Likeness of Being Digital: The Persistence of Nonverbal Social Norms in Online Virtual Environments" in *CyberPsychology & Behavior*, February 2007, Vol. 10 Issue 1

[6] e.g. native speakers delivering conversation using instant messaging,

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Intermediate information layer. The use of the SKOS ontology to create information about e-resources provided by the public administration

Wojciech Górka, MSc
Research and Development Centre for
Electrical Engineering and Automation in Mining EMAG
ul. Leopolda 31, 40-189 Katowice, Poland
Email: wgorka@emag.pl

Michał Socha, MSc
Research and Development Centre for
Electrical Engineering and Automation in Mining EMAG
ul. Leopolda 31, 40-189 Katowice, Poland
Email: msocha@emag.pl

Adam Piasecki, MSc
Research and Development Centre for
Electrical Engineering and Automation in Mining EMAG
ul. Leopolda 31, 40-189 Katowice, Poland
Email: apiasecki@emag.pl

Jakub Gańko, MA
Institute of Innovations and Information Society
ul. Al. Jerozolimskie 123 A, 02-017 Warszawa, Poland
Email: j.ganko@insi.pl

***Abstract*—Currently, the issue of information search is based on processing a large number of documents, indexing their contents, and then evaluating the level of their adaptation to the question asked by the user. The development of the web allows to offer certain on-line services which make it possible to shop, book tickets or deal with public-administration issues. The objective of the WKUP system (Virtual Consultant of Public Services) is to assist the user in the process of searching and selecting services. The system gives a possibility of natural language communication in the first stage of interaction. This functionality has been achieved by means of the SKOS ontology. The article presents a general outline of the WKUP system architecture and the functioning of the search engine which interprets the user's natural-language questions semantically. The article describes the use of the SKOS ontology in the applied answers searching algorithm.**

## I. Introduction

The objective of the WKUP project is to develop a personalized information system which will carry out public administration services with the use of the Virtual Consultant of Public Services (WKUP) based on semantic techniques.

Administration services which can be offered via the Internet are introduced into the public administration step by step, making the citizens' lives easier. These services usually reflect certain procedures and regulations which govern the work of public institutions. On the other hand, the citizen usually uses these services with respect to a larger scale issue he/she wants to settle. In this situation it is necessary to develop a tool which would enable to identify the user's need—the life case, and then guide the user through invoked web services provided by the public administration so that the situation could be dealt with in a complex manner.

The role of WKUP will be to identify the user's issue—life case, give him/her necessary information about that issue, find a relevant public service (or several services), and then to guide the user through the process of complementing the information indispensable to execute the service.

Searching out an adequate process will be done through a preliminary analysis of the user's question and then specifying the issue during the dialogue with the user carried out (to as much extend as possible) in a natural language. At a certain stage of the dialogue, the interaction can be based on selecting the options proposed to the user by the system. During the dialogue the system will collect information about the user (the user's profile will be one of the information sources when user will use system again) and the data necessary to execute the selected process. The user's profile is to facilitate the solution of his/her successive life cases. Both the dialogue and the process of complementing information and the user's profile will be carried out with the use of semantic techniques. The selected service, along with the complemented parameters (those that can be complemented at a preliminary stage of service execution and on the basis of legal and administration terms of the service) will be invoked in a government electronic system.

The architecture will consist of four functional layers (Fig. 1). Two layers, i.e.: the natural language processing layer and knowledge layer, will refer to the WKUP user interface functionality, while the processes layer and web services layer are related to the functional range of the Semantic Broker.

The natural language processing layer will consist of two modules—chatterbot and semantic search engine based on the SKOS ontology. The role of the chatterbot will be to make conversation with the user about casual issues (weather etc.) The role of the semantic search engine will be to find out what the users' need is.

The following chapters describe a part of the system related to the semantic search engine.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

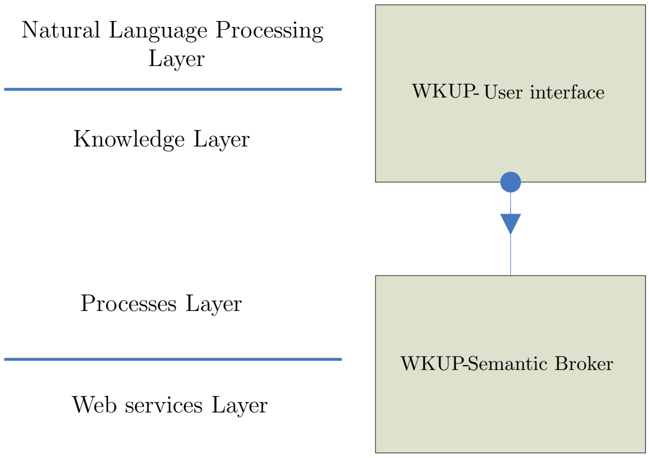


Fig. 1. WKUP architecture

## II. General outline of search issues

There are many solutions in the realm of information search which allow to index the information contents and search for documents based on the contents. The full-text search solutions are mostly based on statistics and there have been many algorithms developed in order to standardize the search results [1]. A relatively new solution are algorithms which allow to cluster the search results [2]. Clusterization introduces the documents selection with respect to areas of interest (a sort of categorization) based on words used in a given text. A category is, to certain extent, a representation of the document contents determined on the basis of the statistics of words used in the document. Examples of such solutions are Vivismo [3] and Carrot2 [4].

One of the full-text search products is the Lucene search software [5]. The software enables to create a properly compressed index and to efficiently search for documents (even concrete places in documents) which are the answer to the question asked by the user. Additonally, Lucene makes it possible to create adapters which allow to browse different types of documents (Microsoft Office documents, XML documents, PDF documents, etc.)

The accuracy of the information search is achieved due to the use of semantic webs solutions [6]. Semantic webs allow to describe information in a formal way and to introduce interdependencies between particular pieces of information. This way the information search is broader. The use of semantic webs will allow the search tools developers to design new–quality products. The search tools, equipped with the knowledge about the concepts hierarchy and their interdependencies, will make an impression of intelligent software. Such knowledge allows to search not only for the key words given by the user but also for the related concepts, and shows how this relation is made. The example are synonims of terms given by the user, semantic relations whole-part or other relations between words.

Irrespective of the development of information technologies there are works caried out in the realm of text corpuses[1], which enable to determine, among others, dependencies between words and the frequency of their occurrence in texts [7]. Such works allow to create word nets (WordNet [8]). The works on the word net for the English language have been carried out since 1985. The works on other European languages (Czech, Danish, German, Spanish, Italian, French, Estonian) were carried out between 1996-1999 within the EuroWordNet project [9]. In Poland the works have been conducted within the plWordNet project [10]. Constructing a word net is done automatically, to a certain extent, thanks to the use of the Polish text corpus. The data from word nets, actually—relations between words, can be used to associate the words which appear in the indexed texts. This way it is possible for the user to find documents on the basis of the question in which the key words included in the document have not been used directly. Thus this solution is similar to proposals derived from the semantic webs concept.

In the realm of information search it is possible to determine the qualities of systems whose objective is to answer the questions. An example is the AnswerBus system [11] based on the knowledge indexed by Internet search tools. The search results are interpreted in an adequate way so that the information looked for by the user could be extracted from the document found by the search tool.

Another interesting solution is the PowerSet search tool [12]. The objective of the tool is to answer the user's questions on the basis of resources in the Wikipedia service. The tool operates on the basis of structures which enable to determine the question context, to select answers into certain thematic categories, and to find related concepts.

## III. Motivation of the execution

Full-text search is based on the statistical analysis of words included in the processed documents. The works ”An Introduction to Information Retrieval” [2] and “Term weighting approaches in automatic text retrieval” [1] present different issues related to full-text search tools operations (indexing, compressing the index, analysis of the user's question and the answer to this question). It is worth mentioning that this approach to searching is based on statistical methods and requires plenty of data in order to achieve accurate and appropriate results. A large number of words in a document, as well as a large number of documents, allow to better select the words which are characteristic of the given document—key words. The solutions based on full-text search tools achieve better results in the case of a large number of long-text documents.

[1] A large and structured set of texts (now usually electronically stored and processed). They are used to do statistical analysis, checking occurrences or validating linguistic rules on a specific universe. They are the main knowledge base in corpus linguistics. The analysis and processing of various types of corpuses are also the subject of much work in computational linguistics, speech recognition and machine translation. (*source: Wikipedia*)

Within the WKUP project a different solution was applied.

As it was mentioned before, the objective of the information system which is currently being developed is, among others, to give the user advice on the life case described by the user. Thus the user will ask a question and the system will propose one or a few possible pieces of advice (previously defined in the system). The advice will have a form of short descriptions explaining the operations the user will have to do in order to solve his/her issue (the descriptions have to be readably short and understandable to the user). For example, if the user asks for help because he/she has a stomachache, the advice given should be a message advising the user to see a GP and proposing an appointment via the WKUP system.

The solution can be briefly described as a set of a large number of answers to potential questions of the users - similarly to FAQ lists (Frequently Asked Questions). A potentially large number of the pieces of advice is the encouragement to develop a search tool which proposes a piece of advice (answer) best related to the real-life situation (question).

A small number of words in the text (short contents of the advice) has a negative impact on the efficiency of the document indexing process. The algorithm calculating potential key words for a given document may take into account wrong words due to limited size of the text.

It is also possible to assume that the potential questions range is, to a certain extent, determined. For example, if the system provides information from the field of medicine, the questions asked to the system will be related to illnesses, symptoms or advice connected with the organization of the national health system.

Additionally, it is important to notice that the users who ask questions will not necessarily use the words and terms included in the advice. Associations between the terms used in the question and in the advice may be even more distant than previously described terms associations in semantic webs. The application of semantic techniques will allow the user to use casual words to form the questions which are asked to the domain system comprising specialized vocabulary. The semantic technique allows an average skilled user to make use of the domain system.

Two presented reasons: small size of questions and answers, as well as different ranges of vocabulary used by the user and the information system, are the basis for the solution which allows to interpret the users' questions and to control the results displayed by the search tool.

## IV. Ontologies and SKOS

In order to present the solution we will focus on issues related to technologies applied in the solution development.

Ontology is a branch of philosophy which tries to describe the structure of reality. As understood by philosophy, ontology allows to explain relations between entities, qualities of these entities, etc., so that the reality could be described. In order to "understand" a section of reality, a computer needs the data that describe this reality, i.e. ontologies. The ontologies (as understood by the information technology) and their application are within the interests of the World Wide Web Consortium (W3C). In 1997 a standard was proposed, and as early as in 1999 W3C published the Resource Description Framework (RDF) standard [13]. The standard was complemented in 2004 with the RDF Schema (RDF-S) specification [14].

RDF allows to record triples of concepts. Each triple is a subject-predicate-object expression. Such a way of concepts recording forms a network of definitions (each object can be a subject in a different triple). Fig. 2 features a sample ontology diagram on which the concepts (circles and squares) are depicted along with their relations (arrows with names).

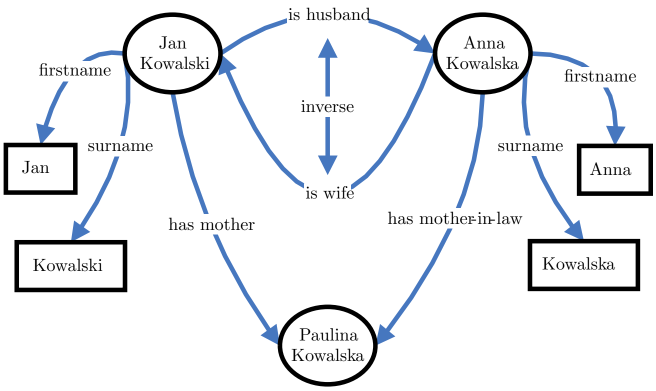


Fig. 2. Sample ontology

RDF-S introduced the possibility to build meta-concepts: classes, sub-classes, features. It also launches a non-standard way of defining the name of the notion (*label*) and its description (*comment*).

The next stage to extend the semantic web standards was to increase the expressiveness of languages intended for ontology recording. W3C published the OWL (Web Ontology Language) standard [15]. The language allows, among others, to express the number of concept sets, to show how one concept belongs to or differs from the other, to identify necessary and sufficient conditions for a given concept. Greater expressiveness of the language allows to verify concepts added to the ontology and to search out certain facts and features indirectly. Additionally, OWL makes it possible to integrate two ontologies by means of associating their identical concepts.

Therefore, ontology description standards allow to describe concepts and the network of links between concepts.

The SKOS specification (Simple Knowledge Organization System) [16], developed and extended under the auspices of W3C, defines an ontology which allows to express the basic structure and contents of concept diagrams, including thesauruses, thematic lists, heading lists, taxonomies, terminologies, glossaries, and other kinds of controlled dictionaries. The specification is divided into three parts: SKOS-Core [17] [18], SKOS-Mapping [19] and SKOS-Extensions [20].

SKOS-Core defines basic concepts and relations which enable to develop concepts and relations between them. SKOS-Mapping introduces relations which allow to describe similaries between concepts created in different ontologies. SKOS-Extensions introduces extentions of the intensity of hierarchical relations from SKOS-Core.

The SKOS ontology assumes that concepts are described by elements linked by means of the *subClassOf* relation with the *Concept* element.

Each concept can be labelled. The SKOS ontology extends the labels that can be used:

- *prefLabel* (chief label of a given concept)
- *altLabel* (auxiliary label, alternative for a given concept)
- *hiddenLabel* (hidden label, e.g. for casual words or other words treated as "hidden" due to other reasons).

The concepts can be linked into hierarchies by means of *broader* and *narrower* relations. The SKOS-Extensions specification introduces extra semantics of hierarchy relations, among others by the following relations:

- *broaderInstantive*/*narrowerInstantive* (express context hierarchies—instances, e.g. Dog and Azorek[2]).
- *relatedPartOf*/*relatedHasPart* (express the whole-part semantics, e.g. Car and Wheel).

The SKOS ontology also provides the class definition which describes a set of concepts—*Collection*. Such a set can help to manage the ontology and facilitate its edition by grouping concepts of similar meanings. Possible ways to use the structures of concepts built on the basis of the SKOS ontology were described in use cases [21]. What is derived from these use cases is, among others, the application of SKOS to the following:

- to order and formalize the concepts used in a given domain, to search—on the basis on the concepts and a part of relations between them—for resources assigned to the concepts,
- to search for information in different languages (thanks to an easy method of translating labels in the ontology with an unchanged relation structure),
- to label press articles, TV programmes, etc. with key words from a thesaurus recorded in accordance with the SKOS ontology.

The above objectives of the SKOS ontology satisfy, to a high degree, the requirements of the search tool in the WKUP system. Therefore a decision was made to apply this ontology. The application was justified by the possibility to provide the tool with a wide and, at the same time, precise "understanding" of concepts. Thanks to semantics it is possible to record the relations between concepts which, in turn, allows to better interpret the questions.

## V. Performance method

The use of the SKOS ontology in the WKUP system consists of two stages: edition and production (search tool operations). Fig. 3 presents the way of using the concepts, defined in accordance with the SKOS ontology, with a view to search for certain resourses—data—related to these concepts.

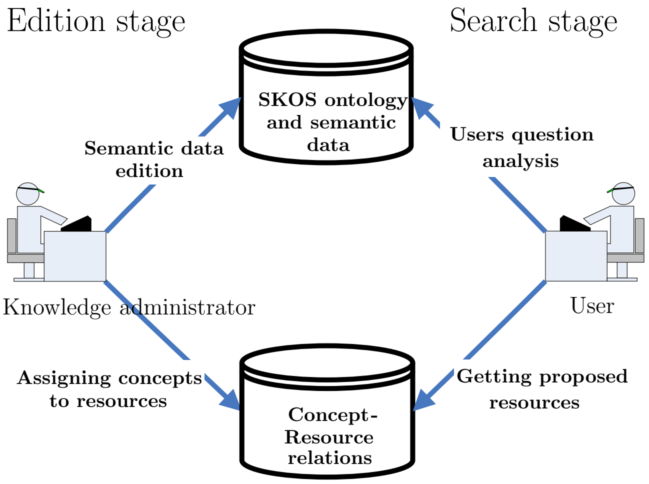

Edition stage
Search stage
SKOS ontology and semantic data
Semantic data edition
Users question analysis
Knowledge administrator
User
Assigning concepts to resources
Getting proposed resources
Concept-Resource relations

Fig. 3. The use of concepts defined in accordance with the SKOS ontology in the search process

At the edition state (before the system starts) the administrator defines concepts and their mutual relations. Then he/she creates relations of the defined concepts with the data which are to be searched for. The ontologies defined in this manner are used at the search stage (production operations of the system). The user's question is analyzed based on the used concepts. The identified concepts are processed. On the basis of mutual relations between concepts, the best fitting answers of the system are found—the resources the user is looking for.

The analysis algorithm of the user's question was divided into successive stages. The first stage is "cleaning" the user's question from redundant non-alphanumeric signs as well as lemmatization of particular words in the sentence. For the statement prepared in such a way, at the next stage the best-fit concepts are searched for based on their labels (relations *prefLabel*, *altLabel* and *hiddenLabel*). In the case when the found concepts are not related to the resources, the relations *broaderInstantive*, *broader* and *relatedPartOf* are used in order to search the web for the concepts which have certain resources assigned. This allows to find the concepts whose meaning is broader than the meaning of the concepts used in the sentence.

The *related* relation is treated in a special way. Thanks to the *related* relation, several concepts which lead to the same resource make the resource "stronger" by assigning a higher searching priority to it. This way it is possible to model the relations between concepts derived from the knowledge about the specifics of the given domain for which the concepts are modelled. The last stage of the sentence analysis is the use of information about the words location with respect to one another in the user's sentence. The words which are closer to one another and point at the same resource simultaneously raise the priority of the found resource. This results from the prerequisite that, usually, the words which determine the same object are located close to one another in the sentence.

Such analysis allows to present the found resources to the user according to the assigned search ranking.

[2] Popular dog name in Poland.

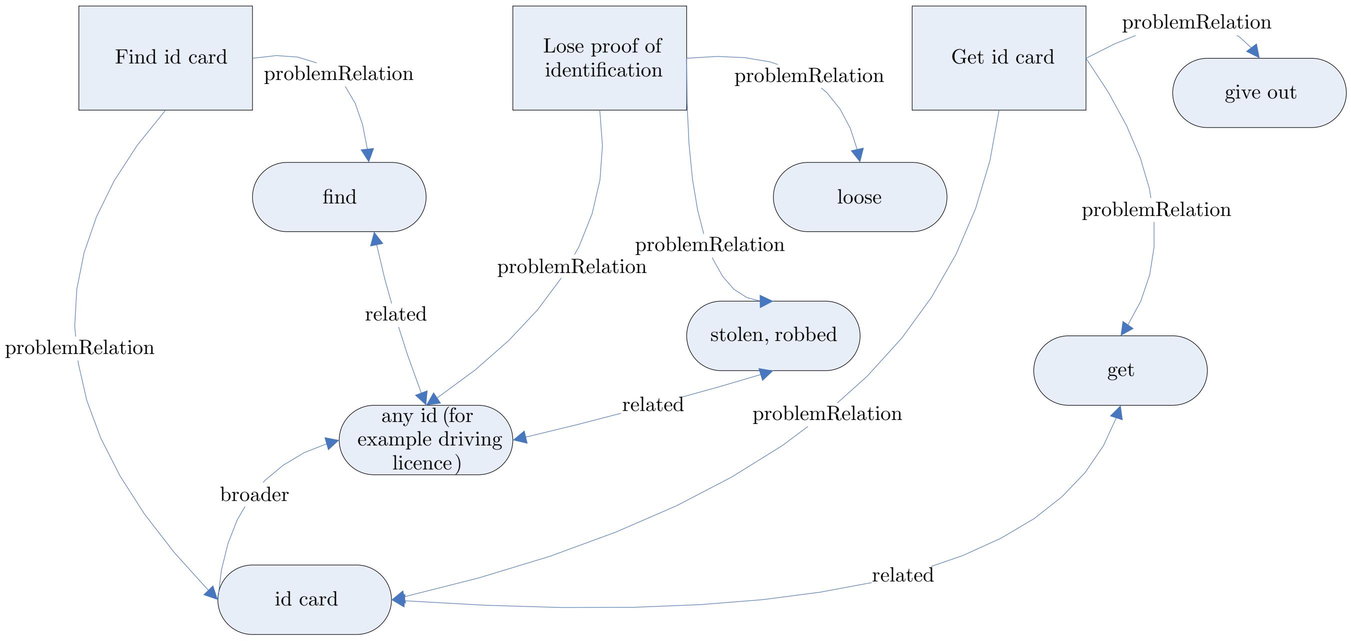


Fig. 4. Sample SKOS structure and its relation to the resources to be searched for

Fig. 4 features a sample SKOS concepts structure and its relation to resources that are to be searched for. Three issues (real life situations) have been defined: finding an ID, losing an ID and getting a new ID. Additionally, the following concepts have been defined: finding, loss, theft, getting and issuing. The *related* relations allow to "strengthen" certain relations other than *broader* and *relatedPartOf*.

Building a net of concepts and assigning resources to the concepts allow to model the system answers to the user's questions. This way the data administrator, who defines the system answer by himself/herself, has a clear picture of the system behaviour with respect to a given class of questions. Such a solution is more deterministic than full-text search tools which operate on the basis of statistical methods only. Additionally, to improve the data administratord's operations in the system, the mechanisms were introduced which function in traditional search tools solutions, but at the edition stage of the ontology. Thus the possibility of automatic collection of concepts from the indexed elements (descriptions of life cases) was applied, and.the process of assigning the concepts to life cases was automatized. In order to perform this task, the algorithm was used to calculate normalized words priorities for documents (*dt* indicator) [2]. The algorithm allows to calculate the adequacy ranking of a given word for the indicated life case. Therefore the work with the tool can start from automatic indexing of life cases and then proceed to successive introduction of revisions by means of successive introduction of relations between concepts, changing labels and their classification (pref, alt, hidden), etc.

## VI. Conclusions

The presented solution is a proposal to solve a certain issue related to information search. It seems that the solution can improve the search in resources which are limited in terms of the number of indexed documents, and in the situation in which it is assumed that the users will ask "questions" to the search tool. The solution appears especially adequate in the case of the so called FAQ lists. They define ready answers to certain questions and, more importantly, the questions are usually relatively short. In such cases full-text search tools can have problems to properly index the contents.

The solution is at the prototype stage now and its operations have not been checked in practice yet. On the basis of the conducted tests it seems, however, that the efficiency of the search tool operations depends mainly on a well constructed ontology. Therefore the ontology is the key element which affects the functioning of the system.

Practical results of the search tool operations and the drawn conclusions will be the topic of the next publication.

## References

[1] Salton G., Buckley C. , "Term weighting approaches in automatic text retrieval. Information Processing and Management 32", pp. 431–443. Technical Report TR87-881, Department of Computer Science, Cornell University, 1987
[2] Manning C.D., Raghavan P., Schütze H., "An Introduction to Information Retrieval," Cambridge UP, Draft of July 1, 2007
[3] Vivismo, http://vivisimo.com
[4] Carrot2, http://www.carrot2.org
[5] Apache Lucene, http://lucene.apache.org, http://www.w3.org/2001/sw/Europe/reports/thes/1.0/guide/20040504/
[6] Semantic Web, http://www.w3.org/2001/sw/
[7] Przepiórkowski A., "The Potential of The IPI PAN Corpus", Institute of Computer Science, Polish Academy of Science, Warsaw
[8] WordNet, http://wordnet.princeton.edu

[9] EuroWordNet, http://www.illc.uva.nl/EuroWordNet
[10] Polski WordNet, http://www.plwordnet.pwr.wroc.pl/main
[11] AnswerBus, http://www.answerbus.com/index.shtml
[12] PowerSet, http://www.powerset.com
[13] W3C, Resource Description Framework, http://www.w3.org/RDF
[14] W3C, Resource Description Framework Schema, http://www.w3.org/TR/rdf-schema
[15] W3C, OWL Web Ontology Language, http://www.w3.org/TR/owl-features
[16] SKOS, Simple Knowledge Organisation System, http://www.w3.org/2004/02/skos
[17] SKOS Core Guide, http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102
[18] SKOS Core, http://www.w3.org/2004/02/skos/core.rdf
[19] SKOS Mapping, http://www.w3.org/2004/02/skos/mapping.rdf
[20] SKOS Extensions, http://www.w3.org/2004/02/skos/extensions.rdf
[21] SKOS UseCase, http://www.w3.org/TR/2007/WD-skos-ucr-20070516/

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 443 – 449

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Security Perception in E-commerce: Conflict between Customer and Organizational Perspectives

Mohanad Halaweh, Christine Fidler
School of Computing
De Montfort University
Leicester, UK
Email: {Mohanad, cf} @dmu.ac.uk

***Abstract*—Security is one of the principal and continuing concerns that restrict customers and organizations engaging with e-commerce. The aim of this paper is to explore the perception of security in e-commerce B2C and C2C websites from both customer and organisational perspectives. It explores factors that influence customers' perceptions of security. It also highlights conflicts between customer concerns with respect to security and those of an organization; existing research has not highlighted this issue greatly. This research provides a better understanding of customer needs and priorities on the subject of security, and enriches the currently available security perception literature, by providing new insights from empirical research conducted in Jordan. A qualitative research approach was adopted, since the research seeks an understanding of human (i.e., customer and organisational employee) perceptions.**

## I. Introduction

SECURITY is the challenge and the main problem for successful e-commerce implementation, as stated by many researchers [1]-[6]. However, there is wide agreement between academic researchers that security is not only a technical challenge; rather it involves managerial, organizational and human dimensions to be more effective [7]-[12]. Therefore, understanding (and acting upon) the customer's perception of security is vital to successful e-commerce interactions, because even when a company uses the best technical solutions that provide full security, without the underlying perception and awareness from customers that their particular website is secure, then these technical solutions may mean nothing. Salisbury [13, p.2] defined security perception as "…the extent to which one believes that the Web is secure for transmitting sensitive information…" (e.g. credit card details), where the meaning of security is subjective and which can therefore vary from one person to the next. The target sample of this research, which seeks to understand security perception from both customer and organisational perspectives, is taken from selected Jordanian organizations and Internet users. The reason for choosing the Jordanian context is justifiable, as most existing research conducted in Jordan confirms the security concern in e-commerce and Internet banking, without exploring the issue in depth [14]-[18]. This barrier (i.e. security) makes both Jordanian organizations and individual customers hesitant to participate in e-commerce transactions, and thus reducing the growth of e-commerce. Therefore, security in e-commerce is a vital area of research, both in general and for Jordan in particular. This research will be the first of its nature in Jordan, focusing on security in these applications from both customer and organizational perspectives. It specifically addresses Business-to-Customer (B2C) and Customer-to-Customer (C2C) e-commerce websites.

## II. Literature Review

In the literature, most security research that is relevant to e-commerce within the IS domain, focuses either on the organization (including technical implementations), or on the customer. In particular, these studies identify systematic processes and factors that need to be considered when implementing a secure e-commerce application from the organizational perspective [19]-[24]. Other research have investigated customers' perceptions and beliefs about security controls and features in e-commerce [25]-[29]. Little research has been conducted which investigates the customer and the organization jointly as a single phenomenon with respect to e-commerce security. This "holistic" view leads to research insight that enables organizations to use certain security solutions that are wholly aligned with customer's objectives and perceptions, thereby reducing the gap between the technology utilized and solutions implemented by organizations, on the one hand, and that being perceived by customers, on the other.

With the existing literature, several factors have been identified as having influence on the customer's perception of security, such as attitude toward security, user's knowledge and experience of security features, ease of use of the interface and presentation of the website, presence of third party security seals such as Verisign and of third party privacy seals like TRUSTe, presence of SSL encryption as indicated by a small padlock, presence of https in the address bar, presence of a security and privacy policy, and the electronic receipt and acknowledgement of the process [25]-[30].

## III. Research Method

A qualitative approach which is suggested by [31] is adopted. Qualitative research is subjective in nature, and involves examining and reflecting on meanings and perceptions of individuals in order to gain an understanding of so-

978-83-60810-14-9/08/$25.00 © 2008 IEEE

cial and organizational phenomenon. It assists the researcher in understanding the target phenomenon in depth and in its natural setting. A total of 27 interviews were carried out; 15 with customers, and 12 with organizations' business managers and IT staff. The questions that were used for asking customers were open, and focused on exploring their perception of security and how they would check to see if a certain website was secure or not. On the other hand, the questions posed of organizational staff focused on identifying their perception of, and viewpoint on, customers' concerns of security and what issues customers needed to know for distinguishing between a secure website from a non-secure one. During the presentation of the results that emerged from the fieldwork, the researchers provide several quotations from the interview transcripts (presented in italics during the results narrative), in order to show how the "story" derived from this research is relevant and grounded on the meanings that were assigned by the participants themselves (as understood by the researchers).

## IV. Research Findings

This section presents the findings regarding Jordanian customer e-commerce security perceptions, from both customer and organizational (through the eyes of selected technical and managerial employees) perspectives.

### A. The Customer perspective

Customer answers with regard to how to check security of a website and what criteria to consider when doing so can be categorized into those referring to tangible features and those referring to intangible features. Tangible features are technological security features on the website that can be checked by users visiting the website, such as https, padlocks, security certificates and security symbols, while intangible ones are not seen on a website yet the user needs to understood or have knowledge of them. They are affected by society in terms of communication and the environment where the customer lives and what they hear from others, as well as their past experience, such as whether the website is well-known and reputable. The perception of the intangible features is constructed by informal word-of-mouth communication between people.

Tangible features need to be understood and checked by the customer on a website, rather than captured through social discourse between people; understanding is gained by having knowledge and experience of these features: for example, in the case of security certificates, a customer needs to know what it means to have one, and how s/he can check to see if it has expired or not.

Some customers indicated that tangible indicators meant very little to them, and much less that the intangible indicators. When considering Website security, the intangible issues surrounding that website were given priority, and only after these had been considered might the tangible ones be checked. For example, the following participant indicated that the presence of a padlock on a website provides him with an indicator of security, but that he does not rely solely on that indicator; rather, he relies more heavily on other people's experience and commendation of the website (an intangible factor).

> *In fact this depends on what people say, for example, I heard that the padlock in the bottom of the page means this website is secure… but the main thing for me is what other people say because they have had experimentations with these websites before me.*

Some respondents indicated that the presence of details about security features on the website and information about website policies, besides the interface design, would make them feel that it is more secure. Examples from the participants' responses are:

> *I mean sometimes the website provides you information that makes you feel a sense that this website is secure and also through transferring between Internet pages, step by step, until arriving at the confirmation page. This gives me feeling that they are serious and secure.*

> *If the website shows the customer a brief description of what security issues they should be aware of and an understanding, then this makes the customer more trusting of the website.*

*In fact this depends on what people say, for example, I heard that the padlock in the bottom of the page means this website is secure… but the main thing for me is what other people say because they have had experimentations with these websites before me.*

Some respondents indicated that the presence of details about security features on the website and information about website policies, besides the interface design, would make them feel that it is more secure. Examples from the participants' responses are:

*I mean sometimes the website provides you information that makes you feel a sense that this website is secure and also through transferring between Internet pages, step by step, until arriving at the confirmation page. This gives me feeling that they are serious and secure.*

*If the website shows the customer a brief description of what security issues they should be aware of and an understanding, then this makes the customer more trusting of the website.*

> *I read their policy and all information that was relevant to the website if I have doubts about this website…*

Participants reported that the availability of a third party, who is neutral and international, can act as an intermediary and be accepted by all parties, thereby guaranteeing that security is provided, making them perceive that a website is secure and consequently enabling their engagement in e-commerce. For example, one participant suggested:

> *I think if there is a security company that is recognized internationally, then this shows that they have a list of websites they registered, as well as mentioning they give these websites a reference number, then the customer once entering a certain website can check this reference number or the brand name of this company in the security company's list…not merely a stamp, rather a reference number… this method makes buying over the internet secure.*

> *Well, some websites mention that they have secure payment but this does not mean indeed it is secure unless it is provided by a third party.*

Several participants perceived that if a site is well known then it must be secure, when they were asked how they would check that a certain website was secure or not. Some participants referred to websites that were trusted by others, and relied on the rating scheme that was provided via that website. For example:

> *As I told you the famous company provides security …. I suppose they do that because they respect their customer. Besides, it shows the security policy on the website and details regarding freight and delivery.*

> *I think the issue here depends on the website, I mean if the website is well known and rated by users, then that is secure and includes an actual address and telephone number then this site is secure.....*

> *If I find the payment via PayPal, then I complete the transaction without any reluctance because it is a well known worldwide company. The reputation of feedback on a website and the rating by customers, and their experience with a website gives proof that the website is secure and credible.*

In this last quotation, the participant highlighted the familiarity of an electronic payment service provider such as PayPal, and the influence on security perception that the presence of such a familiar service has. This was also asserted by another participant when he said:

> *I didn't hear any one censure Amazon… you know why - because this website deals with the largest company in the world; like PayPal, it undertakes the security on the website, these websites pay millions for that.*

The reputation of a website was also reported by other participants, and that it is the base upon which customers relied when considering to buy online:

> *I think there is no way to say this website is secure or not, the only thing is the reputation of the company's website.*

> *If I do... I do just from a company I have already dealt with or a company that has a good reputation in Jordan.*

A few participants highlighted the significance of the website's existing known (and typically physical) identity, such as that of the banks and telecommunications companies in Jordan; participants appear not to use websites which are anonymous and have no real physical location:

> *I trust only the bank …. Suppose if anything happens then I can go to them and refer to them since they have a physical place.*

> *In fact, I was a customer of this bank for long time so I discarded the fact that the bank would steal or trick me. But there are some instructions, I should know them as a customer to protect myself from any person. For example, the first time when I logged onto the system (website page) it forced me to change the password after six months, I did that but later on I tried to enter it... I forgot the new password I tried many times and then the system asked me to contact my branch to activate my password, because I entered the incorrect password more than three times…… all of these processes are in order to protect me so they are very concerned on security*

The last quotation also showed that, from this participant's perspective, the security of his online account is addressed by providing a strong password procedure. This reinforced his perception that the bank is working to provide on-line security for its customers.

Some participants reported that the characteristics of the company (e.g. respected and large size) would lead them to feel that it supports and provides secure website access.

> *The company that respects its customers, is well known and especially larger ones, implicitly provides the required conditions in order to complete the transaction in secure way.*

One of the participants has not yet bought online but in his opinion there are definitely secure websites. He stated:

> *I hear that there are many people purchasing over the internet and some people buy and sell shares as well, so certainly it is secure as there are people doing it, otherwise why do they buy and sell if it is not secure.*

Table I. summarizes all the tangible and intangible indicators of security from a customer's viewpoint that were derived from the fieldwork, many of which have been touched upon within the preceding discussion (others were not discussed due to paper length limitations, and those that were provide more sufficient evidence of the process of research narrative development). On close inspection, it may appear that several of the intangible indicators appear to be identical: for example, famous, well-known and recognized could be considered to be synonyms. However, the researcher has kept to the customer's exact phrases rather than presenting (and thereby enforcing) his own interpretation of the words used.

Table II.
Tangible and intangible security features on e-commerce websites from customers' perspectives

| Security features in e-commerce website | Categorizing of security features |
|---|---|
| Padlock | Tangible |
| Security certificate | Tangible |
| Transferring between interfaces of the website<br>Website presentation | Tangible |
| Security policy | Tangible |
| Acknowledgment via email | Tangible |
| Third party symbols | Tangible |
| Physical address , telephone # and email | Tangible |
| Brief description of the security issues that the customer should be aware of on the website | Tangible |
| Known identity (company has physical building, i.e. Bank) | Intangible |
| Support password system | Tangible |
| Well-known electronic payment gateway such as PayPal | Tangible-Intangible |
| Famous brand/company | Intangible |
| International | Intangible |
| Recognized | Intangible |
| Trusted | Intangible |
| Well-known | Intangible |
| Formal website | Intangible |
| Respected company, large size | Intangible |
| Reputable | Intangible |
| Well-rated | Intangible |

### *B. The organizational perspective*

Some respondents indicted their impression in general about a customer's acceptance and engagement in e-commerce. For example, one participant believed that customers have a generally negative attitude towards online shopping, and that there is no trust and transparency between the customer and merchant:

> *In fact, and by my experience with an e-commerce website for several months, I arrived at an unbelievable fact that Jordanian citizens and Arabic customers in general don't believe or trust shopping online, they think no transparency is provided by the websites. For example traditionally, when the customer buys a computer from a store, he faces a problem if he wants to fix his computer, he is countered with violation of the deal by the merchant, and he is always the weakest party and will ultimately carry the cost of fixing the produce. So, how do we persuade the customer buying online that whilst he does not see anything tangible in front of him, and is not able to touch it with his hands, where he already had faced problems with a physical store ... he needs guarantees.... really, where the detailed information that is provided by the first page on the website is not sufficient to convince him… briefly, the trust between the customer and merchant is nonexistent as it is not between the customer and Arabic governments.*

In contrast, however, another participant was optimistic by showing the achievement of her company, and the degree of online acceptance from customers. She commented that customers nowadays are more aware and have greater propensity to accept online trading given their experiences of using ATMs and the generally wider availability of credit cards.

> *We applied an electronic ticketing system on our website which was an important factor in enabling our business, as a result it has become easy for customers to book a ticket and pay online from anywhere… people are accepting that they are ready more than you think, there was minor rejection but in general it was accepted smoothly by our customers…really, we are surprised how people are ready to accept it…… Customers nowadays become used to an ATM and it is not big deal, most of people have a credit card… people are more developed than in previous years.*

She asserted that her company focuses on strong customer support to allay and respond to customer concerns, by saying:

> *The customer viewpoint is considered and we have a customer service centre that is responsible for customer's enquires, claims and problems, and we take on their feedback which is important for us.*

Another participant pointed out how his company's concern was about the customer in respect of the website design. Here, the respondent correlated ease of use in the website with a feeling of security, but immediately went on to say that this, in itself, is not really security.

> *The user's viewpoint is necessary for us, providing a website that is easy to deal with, friendly, motivates the customer to use it, which makes him feel it is secure to some extent...but not exactly secure.*

The next participant showed how his company considers the customer viewpoint, and what it does to provide secure online transactions. In his viewpoint, a simple and true (i.e., product exactly matches what the customer expects) transaction with the customer makes him/her feel that the website is secure, and this makes the customer experiment and eventually become a repeat user of the website.

> *In fact, the facilities and services that we provide and our security is not just talk, but the fact that when a customer enters our website and obtains the product that he wants with the same specifications, this happens without any complexity and is easy. This makes them come again because they found our sincerity of treatment. Now we have more than 10,000 active users who buy and sell over our website. Those, once they have tried and succeeded, and have found it is secure, they will then become one of our customers and users of our website….it is just the first experimentation.*

He continued by pointing out that the company website also has a forum where sellers and buyers can chat, get to know each other, share opinions, provide suggestions, report transaction problems, recommend certain sellers and certain products, provide feedback, and request support from the website. In addition, the website provides a rating system which makes customers feel that the website has greater credibility.

> *One of the important things that makes customers feel that our website is secure and credible is the rating system which indicates positive and negative ranking for buyers and seller, and the best buyers and sellers……We have a forum on our website and we have seen for example one customer ask a question and another customer tell him to refer to our policy, the clause number#. If the customer faces any problem, we resolve it within 24 hours, the nature of our website is that easy to deal with. It is so that it makes customers feel happy and confident and in control over the work on our website, it is not complicated.*

He added that their website is 100% secure and they guide their customers on the first page to check the padlock.

> *We put on our website 'secure 100%', we carry the responsibility for that, there are websites that say that they are secure 100 % but they are not secure and just talk rubbish. On our website, we also provide customers with an explanation regarding the security privacy policy*

And on that the website, the following instructions are found :

> *Look for the item with this icon* 🔒
>
> *This means that the auctions displaying on it are more secure.*

In contrast, another participant stated that customer concerns are addressed solely by the services provided on the website:

> *The customer viewpoint is considered at service level what he would like to see and what he wouldn't like.*

Some of the technical staff involved in the development and maintenance of organizational websites did not consider the checking of tangible indicators as a sufficient mechanism for determining the security, or otherwise, of a website. This is firstly because technical staff appeared to be unconvinced that tangible indicators provide real security; websites are hacked despite the presence of these indicators, so customers could be led into a false sense of security by relying solely on them. Secondly, these indicators assure customers of the organization's honesty, for example, by using security certificates as an indicator that the website is guaranteed by a third party, and thus that the website is secure, but this is no assurance that it cannot be breached by hackers.

> *It is difficult to consider that a website is secure or not even if you are professional, no way to say 100 % secure, and using security certificates just means that you are not lying to your customers by doing your responsibility and this is not a guarantee that you are not hacked by hackers.*

These indicators (e.g. security certificate) are thus not sufficient to assure that the website is completely secure. Here, the risk does not come from the website itself but might be from outside parties (e.g. hackers). One participant maintains that there is no way to confirm that their websites are secure or not, even when such websites are very well-known.

> *Frankly, there is no way to judge that a certain website is secure, even though it is Amazon and eBay…. it is reputation and ease of use, the guarantee is the experiment and reputation.*

From the interviews with organizational members, it was also found that naïve customers are sometimes not aware of technological details, such as the meaning of terms like https. Examples from participants reported that security understanding is not an important thing for the customer and that the only concern is others people's experiences or reputation of the website.

> *Some people don't care about security, they don't think is it secure or not, they are just concerned about what other people say if it is well know-company and credible by others, then they use it and trust it.*
>
> *To say this website is secure 100 % means noting for the customer, I think the reputation of the website makes the customer feel it is secure.*
>
> *The main concern for users is the reputation of the company, they are looking for a well-known company, and people here in Jordan deal with national companies like telecommunications because they know them well.*

Thus, from an organizational perspective, customers look for intangible indicators such as reputation, a well known company, and the system of rating by previous customers of the website, in order to assess whether or not a website is sufficiently trustworthy to engage with, regarding online purchases.

## V. Discussion

Based on the findings in the previous section, conflict between the some of the views of organizations (in the eyes of management and technical personnel) as to what customers consider important, and those views of the customers themselves, can be clearly seen. The research findings showed that some of the participating organizations indicated their acknowledgement of customer security concerns by stating them on the first page of the website that it was 100% secure. One participant stated that their website then guides customers to check whether a padlock symbol appears when a transaction is performed, advising them that if it does then the website is secure. This raises the question of whether such advice is sufficient to convince a customer purchasing online or performing online transactions. In essence, while it is important, it may indeed not be sufficient; a responsible company should explain website security to its customers, not merely present a logo or use a short sentence to state that it is secure. Rather, they need to make clear what the padlock means, what technology is used to encrypt the data and what protocols are applied. One participant from an organization pointed out that customers' concerns are considered at the service level, and this should be accepted as a premise and therefore it should be considered that customers are only concerned with the quality of the service provided on the website and whether it is easy to operate. In essence, this can be refuted by the argument that some users know the meaning of security indicators on websites, so that in this context, new thoughts are required; a change of attitude is needed among technical staff, because they tend to underestimate customers' perceptions and their ability to understand what is involved. Thus, organizations exempt themselves from fulfilling their responsibility to educate their customers in issues related to security.

On the other hand, customers' responses have revealed that they do not intensively check tangible security features, being more interested in knowing the identity of the other party; they want to know whether they are dealing with a national company which is well-known, famous and reputable, which are intangible features. If these questions are answered affirmatively, then the customer feels secure. For such customers, security is guaranteed on the basis of the abovementioned features. Consequently, more effort is required by the organizations in this field, namely, to seek strategies to make their websites better known and to boost their reputation. For example, if a customer wants to buy something from Amazon, then does he check whether the site is secure or not? This example would suggest that tangible security features on the website are not essential, but that the customer will decide to buy from the website without checking, simply because he depends on the reputation of this website – and in this case the reputation of the company implies by default that the website provides the required security. This raises another question: does every company which has a good reputation actually guarantee the security of its website? In essence, it can do so only if the company is responsible for the protection of the customer's data from its actions. For example, if a company's website applies the best technology for encryption of customer's data, but then their private data is transferred by the website to another party without the customer's consent, then the violation of security (i.e. confidentiality/privacy requirement) has come from the website itself, despite its supposed reputability; so, the responsibility of the company for security should have two dimensions: protection of data from hackers, and from misuse by the parent organization of the website itself.

Organisational staff indicated that tangible features do not totally guarantee security. The consideration addressed by organisational staff, that there is no way to judge whether a website is secure or not, leads to a reasonable enquiry: if there are no dependable criteria for distinguishing a secure site from an insecure one, what should the customer depend on when purchasing online securely? The justification for this doubt is that while security features (e.g. using SSL, security certificates) of the website may mean that its operator has an honest stance towards its customers, that while their data is encrypted for transmission, that the website's identity is authenticated by a third party and that this, as reported by one participant, means that they do not deceive their customers, and that while the website undertakes to provide secure transactions, none of this means that the company is able to totally guarantee that their site will not be hacked or its security breached. In other words, as another participant stated, it is difficult for even well-known websites to guarantee total security. In essence, this shows how such a significant role is played here by the intangible indictors of security, such as the fame or reputation of the website, which represents the first priority for a customer in deciding whether to buy online. Fame or reputation of the website assures him that the website's operators undertake the responsibility to protect his data. This concurs with the organisation's view that customer concerns are about the reputation of the website, how well known it is, and how it scores on rating schemes, for example. It may be concluded that tangible and intangible security features are both important and need to be checked by customers, who should not depend entirely on one or the other.

Although the core idea of this paper is to investigate security perceptions from the customers' and organizations' perspectives, the researcher has found it is difficult to put some of the participants' responses with respect to trust away, where this terminology was mentioned in their answers. The literature review provides, theoretically and empirically, a set of antecedent factors for trusting e-commerce websites. These factors include the characteristics of the online vendor, third-party certification, the individual's propensity to trust, the influence of perceived risk, perceived security control (i.e. authentication, no-repudiation, confidentiality, privacy concern and data integrity), perceived competence, legal framework, previous experience, perceived credibility, perceived ease of use, perceived privacy, perceived company reputation and willingness to customize products and services, perceived website usefulness, third party recognition, perceived investment, perceived similarity, perceived control, perceived familiarity, and perceived size [32]-[37]. Based on the above, it can be said that perception of security from the customer perspective is determinant on trusting the website, where perceived security is one amongst many other factors that can increase or decrease this trust. Intangible features of security were revealed by the customers, mentioned in Table I., such as reputation, well-known or perceived familiarity of the website, also increase or decrease in belief that whatever certain website is secure or not, even though these features are similar to some of antecedent factors for trust, such as reputation. As a result, this paper extends current literature to show that these factors also influence customer perception of security which is similar to the influence on customers trusting a website.

## VI. Conclusion

This paper has provided a valuable contribution, by providing insight into the customer's perception of e-commerce security. It has clearly identified that both tangible and intangible features play a major role in the customer perception and judgement of the security of a website. It has highlighted and discussed differences between customer and organizational viewpoints of customer e-commerce security perception, and has delivered guidelines for organizations such as the taking on of the responsibility to educate customers towards security features (e.g. security certificate), what these mean and how to check that the website has these. By taking and achieving this responsibility in protecting customer's data in line with making promotional strategies to make their website more well-known and used, its reputation will increase.

In addition, the paper extends the existing body of knowledge by providing evidence that some factors that influence customer's perception of security are similar to that which makes them trust the website as reported in reviewed literature. Therefore, and to provide sound evidence, this stimulates future research which can address the relation between security and trust, and which can identify, by empirical research, whether the factors that influence customers' security perceptions are the same as those that influence trust (or indeed where they differ). This could be achieved by investigating the perceptions on the two issues together based on the same respondent set.

## References

[1] A. Annie, and A. Earp, "Strategies for developing policies and requirements for secure electronic commerce systems," *1st Workshop on Security and Privacy in E-Commerce at CCS2000*, Athens, Greece, 2000.

[2] S. Hawkins, D. C. Yen, D. C. Chouo, "Awareness and challenges of internet security," *Information Management & Computer Security*, vol. 8, no. 3, pp. 131-143, 2000.

[3] L. Labuschagnce, J.H.P Eloff, "Electronic commerce: the information security challenge," *Information Management & Computer Security*, vol. 8, no. 3, pp. 154-157, 2002.

[4] A. Albuquerque, A. Belchior. "E-Commerce websites: a qualitative evaluation.," *The Eleventh International World Wide Web Conference,* Hawaii**,** 2002.

[5] S. Kesh, S. Ramanujan and S. Nerur, "A framework for analyzing e-commerce security," *Information Management & Computer Security*, vol. 10, no. 4, pp. 149-158, 2002.

[6] S. K. Katsikas, J., Lopez and G. Pernul, "Trust, Privacy and Security in e-business: requirements and solutions," in *Proc. of the 10th Panhellenic Conference on Informatics (PCI'2005)*, Volos, Greece, 2005, pp. 548-558.

[7] F. Bjorck, "Institutional theory: a new perspective for research into IS/IT security in organizations," *Proceedings of the 37th Hawaii International Conference on System Sciences*, 2004.

[8] Z. Shalhoub, "Trust, privacy, and security in electronic business: the case of the GCC countries," *Information Management & Computer Security*, vol. 14, no. 3, pp. 270-283, 2006.

[9] B. Von Solms, "Information security–A multidimensional Discipline," *Computers & Security*, vol. 20, no. 6, pp. 504-508, 2001.

[10] C. Bruce Ho and S. Chang, "Organizational factors to the effectiveness of implementing information security management," *Information Management & Computer Security*, vol. 106, no. 3, pp. 345-36, 2006.

[11] G. Dhillson and J. Backhouse, "Current direction in IS security research: towards socio-organizational perspective," *Information Systems Journa*l, vol. 11, no. 2, pp. 127-153, 2001.

[12] J. Elofe and M. Elofe, "Information security management – a new Paradigm," *Proceedings of SAICSIT*, 2003, pp. 130-136.

[13] W. Salisbury, R. Pearson, A. Pearson and D. Miller, "Perceived security and world wide web purchase intention," *Industrial Management & Data Systems*, vol. 101, no. 4, pp. 165-176, 2001.

[14] N. Al-Qirim "The adoption and diffusion of e-commerce in developing countries: the case of an NGO in Jordan," *Information Technology for Development*, vol.13, no. 2, pp. 107 –131, 2007.

[15] S. Alsmadi,. "Consumer attitudes towards online shopping In Jordan: Opportunities and challenges," *The First Forum for Marketing in Arab countries,* Sharjiha, UAE, 2002.

[16] A. A. Al Sukkar and H. Hasan, "Toward a model for the acceptance of internet banking in developing countries," *Information Technology for Development*, vol. 11, no. 4, pp. 381-398, 2005.

[17] M. Sahawneh, "E-commerce: the Jordanian experience," Royal Scientific Society., 2003.

[18] K. M. Titi, "The impact of adoption electronic commerce in small to medium enterprises Jordanian companies," *International conference in e-business and e-learning,* Amman, Jordan, 2005.

[19] K. Knorr and S. Rohrig, "Security requirements of e-Business processes," Towards the E-Society: E-Commerce, E-Business, and E-Government, *First IFIP Conference on E-Commerce, E-Business, EGovernment,* Zurich, Switzerland, 2001, pp. 73-86.

[20] S. Kesh, S. Ramanujan and S. Nerur, "A framework for analyzing e-commerce security," *Information Management & Computer Security*, vol. 10, no. 4, pp. 149-158, 2002.

[21] A. Sengupta, C. Mazumdar and M. Barik, "e-Commerce security – a life cycle approach," *Saddhana*, vol. 30, no. 2 &3, pp. 119–140, 2005.

[22] A. Zuccato, "Holistic security management framework applied in electronic commerce," *Computer and Security* , vol. 26, pp. 256- 265, 2007.

[23] S. Lichtenstein and P. Swatman, "Effective management and policy in e-Business," *Security e-Everything: e-Commerce, e-Government, e-Household, e-Democracy 14 th Bled Electronic Commerce Conference, Bled*, Slovenia, 2001.

[24] J. Rees, S. Bandyopadhayay, and E. Spafford, "Policy Framework for interpreting risk in eCommerce security," *Communications of the ACM,* vol. 46, no.7, 2003.

[25] A. Sharma and W. Yurcik, "A study of e-Filing tax websites contrasting security techniques versus security perception," *Proceedings of the Tenth Americas Conference on Information Systems*, New York, 2004.

[26] C. Turner, M. Zavod and W. Yurcik, "Factors that Affect the Perception of Security and Privacy of E-Commerce Web Sites," *Intl. Conf. on E-Commerce Research (ICECR),* 2001.

[27] S. Singh, "The social dimensions of the security of internet banking," *Journal of Theoretical and Applied Electronic Commerce Research,,* vol. 1, no. 2, pp. 72 – 78, 2006.

[28] M. Yenisey, A. Ozok, and G. Salvendy, "Perceived security determinants in e-commerce among Turkish university students," *Behaviour & Information Technology*, vol. 24, no. 4, pp. 259-274, 2005.

[29] C. Centeno, "Soft Measures to build security in e-Commerce payments and consumer trust," *Communications & Strategies,* vol. 51, 2003.

[30] S. M. Furnell, "Considering the Security Challenges in Consumer-Oriented eCommerce," *The 5th IEEE International Symposium on Signal Processing and Information Technology,* Athens, Greece, 2005, pp. 534-539.

[31] A. Strauss, and J. Corbin, *Basics of qualitative research: grounded theory procedures and techniques*. SAGE Publication, London, 1990.

[32] S. L. Jarvenpaa, N. Tractinsky, and M. Vitale, "Consumer trust in an -internet store," *Information Technology and Management* , vol. 1, no. (1/2), pp. 45–72, 2000.

[33] B. Suh and I. Han, "The impact of customer trust and perception of security control on the acceptance of electronic commerce," *International Journal of Electronic Commerce*, vol. 7, no. 3, pp. 135-161, 2003.

[34] C. M. K Cheung and M.K.O Lee, "An integrative model for consumer trust in internet shopping," *in Proceedings of the European Conference on Information Systems (ECIS)*, Naples, Italy, 2003.

[35] M. Koufaris and W. Hampton-Sosa, "The development of initial trust in an online company by new customers," *Information & Management*, vol. 41, no. 3, pp. 377–397, 2003.

[36] R. Connolly and F. Bannister, "Consumer trust in internet shopping in Ireland: towards the development of amore effective trust measurement instrument," *Journal of Information Technology*, vol. 22, no. 2, pp. 102-118, 2007.

[37] M. Teltzrow, B. Meyer, and H. Lenz, "Multi-channel consumer perceptions," *Journal of Electronic Commerce Research*, vol. 8, no. 1, 2007.

# 4th Workshop on Large Scale Computations on Grids

LARGE Scale Computations on Grids (LaSCoG) Workshop will be organized within the framework of the International Multiconference on Computer Science and Information Technology and co-located with the XXIV Autumn Meeting of the Polish Information Processing Society.

The emerging paradigm for execution of large-scale computations, whether they originate as scientific or engineering applications, or for supporting large data-intensive calculations, is to utilize multiple computers at sites distributed across the Internet. In particular, computational grids are collections of distributed, possibly heterogeneous resources which can be used as ensembles to execute large-scale applications. While the vision of the global computational Grid is extremely appealing, there remains a lot of work on all levels to achieve it. In this context the LaSCoG workshop is envisioned as a forum to promote exchange of ideas and results aimed at addressing complex issues that arise in developing large-scale computations on Grids and running applications on them.

Covered topics include (but are not limited to) Grid-focused aspects of:

- Large-scale algorithms
- Symbolic and numeric computations
- High performance computations for large scale simulations
- Large-scale distributed computations
- Agent-based computing
- Data models for large-scale applications
- Cloud computing
- Security issues for large-scale computations
- Science portals
- Data visualization
- Performance analysis, evaluation and prediction
- Programming models
- Peer-to-peer models and services for scalable grids
- Collaborative science applications
- Business applications
- Data-intensive applications

## International Programme Committee

**Rui Aguiar,** Universidade de Aveiro, Portugal

**Mark Baker,** University of Reading, UK

**Andrej Brodnik,** University of Primorska, Slovenia

**Pasqua D'Ambra,** ICAR-CNR, Italy

**Beniamino Di Martino,** Seconda Universita' di Napoli, Italy

**Prabu Dorairaj,** Wipro Technologies, India

**Salvatore Filippone,** Universita di Roma "Tor Vergata", Italy

**Maria Ganzha,** Systems Research Institute Polish Academy of Science, Poland

**Daniel Grosu,** Wayne State University, USA

**Ching-Hsien Hsu,** Chung Hua University, Taiwan

**Tetsuo Ida,** University of Tsukuba, Japan

**Aneta Karaivanova,** Institute for Parallel Processing, Bulgaria

**Dieter Kranzlmueller,** Johannes Kepler University, Austria

**Anna T. Lawniczak,** University of Guelph, Canada

**Thomas Ludwig,** Heidelberg University, Germany

**Carlo Mastroianni,** ICAR-CNR, Italy

**John P. Morrison,** University of Cork, Ireland

**Richard Olejnik,** CNRS & University of Lille I, France

**Kalim Qureshi,** Math and Computer Science Department, Kuwait University, Kuwait

**Erich Schikuta,** University of Vienna, Austria

**Ha Yoon Song,** Hongik University, Korea

**Przemyslaw Stpiczynski,** Maria Curie-Sklodowska University, Poland

**Pavel Telegin,** Joint Supercomputing Center, Russia

**Nam Thoai,** Ho Chi Minh City University of Technology, Vietnam

**Marek Tudruj,** Institute of Computer Science Polish Academy of Sciences & Polish-Japanese Institute of Information Technology, Poland

**Chao-Tung Yang,** Tunghai University, Taiwan

**Laurence T. Yang,** St Francis Xavier University, Canada

**Baomin Xu,** Beijing Jiaotong University, China

## Organizing Committee

**Marcin Paprzycki,** Systems Research Institute Polish Academy of Sciences, Poland

**Dana Petcu,** Western University of Timisoara, Romania

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 453–457

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Unicore 6 as a Platform for Desktop Grid

Jakub Jurkiewicz*‡, Krzysztof Nowiński*, Piotr Bała* †
* Interdisciplinary Center for Mathematical and Computational Modelling, University of Warsaw
Pawińskiego 5a, 02-106 Warsaw, Poland
†Faculty of Mathematics and Computer Science, Nicolaus Copernicus University
Chopina 12/18, 87-100 Toruń, Poland
‡Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
Banacha 2, 02-907 Warsaw, Poland

***Abstract*—The paper shows a possibility of arranging a desktop grid based on the UNICORE 6 which is a well established grid middleware. The grid consists of a number of PC computers which can communicate with the server in a secure way and perform scheduled computational tasks. The main advantage of this system is ease of deployment, flexibility and ease of integration with the large scale grid. We present here results of a simple performance test as well.**

## I. Introduction

Nowadays the community grid computing becomes more and more popular with a number of architectures available. Boinc [2] package, Condor [3] are good examples here. At the same time we observe rapid development of full featured grid middlewares such as Unicore and Globus Toolkit. Intense works are carried out which aim at connecting desktop grids and full size grid infrastructures. One of the simplest propositions is to create an interface that would allow to run Globus or Unicore jobs on the desktop grid. Condor/G is a good example here. Currently, interfaces that would allow using full size grid nodes for desktop grid are available. Both solutions have one great disadvantage—they make a connection between two different systems and middlewares which causes technical problems. Of course there are some current works which uses this solutions, however they usually mean creating some kind of bridge between middlewares[4][5][6].

In our work we present a new solution: Unicore 6 middleware is used to create a desktop grid. This solution makes the connection of systems really easy, simplifies all problems related with authorisation and authentication and minimises cost of middleware. This, only a very beginning stage of creating a desktop grid middleware proves that Unicore suits well as a middleware for creation of a desktop grid.

## II. Unicore

Unicore is a Java based grid middleware. Early versions of the system (up to Unicore 5) had communication based on the exchange of serialised Java objects. This solution was very fast and easy to implement, but it worked only if both sides of a given communication used the same version of Java. Unicore 5 has been still in use, and one of its main advantages is good separation of the user from the computing system executing his job. This allows connecting computing nodes with completely different architectures. For a certain time now a completely new version of Unicore has been available with the communication based on web-services. It combines simplicity of previous version with portability and independence of Java version from web services. In this paper we refer to Unicore 6 [1].

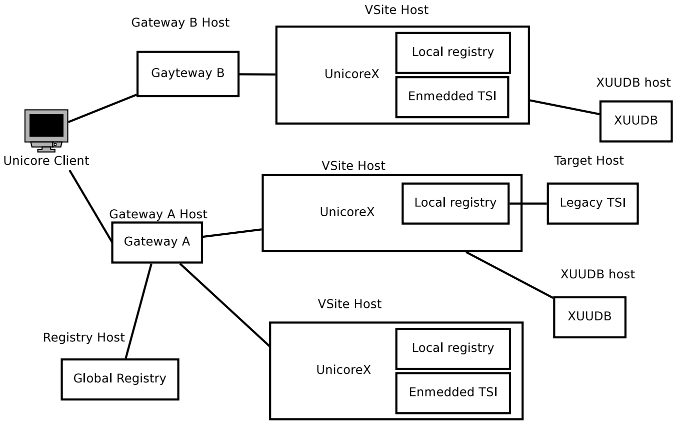


Fig. 1. Example Unicore installation

The Unicore system consists of the following parts:

- Gateway - a module that allows other modules to connect to the grid. It ensures that no unauthenticated user has any access to the protected part of a grid.
- Virtual site (VSite) and a target system (i.e. UnicoreX) - the part of Unicore that is responsible for execution of applications.
- Registry - a holder of information on services used by clients.
- Storage - used for data storage on the grid.
- XUUDB - user database for authentication of users.
- Client - part of Unicore which runs on client's side.

An example Unicore 6 installation is presented on Fig. 1.

A virtual site splits into UnicoreX and Legacy TSI which provides for connecting to target systems built using Unicore 5.

Additionally Unicore has been extended by UVOS system for Virtual Organisations [7]. This extension will allow integrating created desktop grid with the computational grid, as one of possible sites.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

*1) Why Unicore?:* Unicore and Globus Toolkit are two most popular grid middlewares. They both, in their new versions are based on web services. Unicore has two great advantages—it is easy to be configured and run, and it has a simple but very extensible and powerful authentication and authorisation infrastructure which have been based on industry standards such as the X.509 PKI.

## III. Desktop Grid Architecture

The goal of this work is to build, using Unicore modules, an architecture model consisting of:

- computing element (desktop node, potentially unreliable)
- manager node

Current architecture of desktop grid is presented in the Fig. 2.

### A. Computing Element—Node

Computing element which does the most of computations has been built based on gateway and target system—UnicoreX. It uses XUUDB that works as manager for authorisation.

The problem was to minimise the application so that to make it running on the computing element under the target system. An ideal application consists of code that could be downloaded with a job to be executed. Unfortunately, this leads to serious security problems. It is possible only if the application run inside the target system, has the same security as unsecured java applet. This means that the application can not:

- run or read from disk,
- connect to host other then a Desktop Grid Manager,
- use only a secure classloader.

At the early stage of project we still use a disk for keeping logs and keystores, but in the future, after all optimisations we plan to totally disallow the application to use disk for execution. Data for computation is kept on the manager computer and downloaded straight to the memory space of application.

*1) Application Runner:* All jobs submitted to computing element could be divided into three parts:

1) obtaining data from grid storage using the key provided in the job description,
2) doing actual computations using a module described in the job description (Application),
3) sending back the data to the grid storage.

All data is kept in the memory, thus involving no need to use a disk. The private key used for data transferring is encoded into the ASCII string and it could be given to the application as a normal run time argument.

### B. Manager Node

Desktop Grid Manager node software consists of the following elements:

- gateway—which allows accessing to the registry and storage,
- registry,
- grid storage accessible via RBYTEIO,
- XUUDB which is accessible on its own port,
- Desktop Grid Manager.

Additionally, on the manager node we separate the storage space for finished tasks results. It would increase the security, because Desktop Grid Manager is responsible for moving the results there and, after such moving, is the only one who has access to the data. More detailed description of Desktop Grid Manager is presented in III-C.

It is also possible to split Desktop Grid Manager among registry, storage, XUUDB and computations manager. This would allow running jobs from a computer that has no external IP address and/or open ports. Of course, another solution is to create Desktop Grid Manager that works as a service under Unicore.

The architecture we have chosen is very good at this stage of development of desktop grid. It allows us to easily experiment and change parameters of tested architecture.

### C. Desktop Grid Manager

Desktop Grid Manager is a multithreaded application. It uses threads for controlling different aspects of desktop grid work. It allows running a job on a desktop grid and getting results, and is responsible for:

1) checking available nodes in registry,
2) dividing job into sub-tasks and merging results,
3) submitting tasks to a computing element,
4) fetching sub-tasks results,
5) monitoring state of nodes.

Desktop Manager uses separate threads for:

- checking registry—this thread is used for checking if any new computing element has registered, and if there is a new node, thread tries to do the rescheduling.
- checking node—checks if node is still alive, and what is state of computations. If node is down, or if it has finished computations, the thread tries to do rescheduling.
- running manager tasks—some tasks have to be done on the manager server, i.e. dividing job into sub-tasks and merging the results. There is a specially designated thread with its own queue for doing such job. When it finishes a task it tries to do the rescheduling.

*1) Checking Available Nodes in the Registry:* When the owner of a private computer turns on desktop grid infrastructure on his computer, the UnicoreX registers in the Registry located at the Desktop Grid Manager. A thread that checks the registry finds out whenever a new node becomes available and, if so it runs a new thread for checking the node.

*2) Dividing Job Into Subtask and Merging the Results:* When the desktop grid receives a job to do, it divides it into sub-tasks. It is performed by a thread used for running manager tasks. This thread also runs a part that merges results after they are fetched.

*3) Submission of Task to a Computing Element:* If any of the following events happen in the system, such as:

- node is up or node is down,
- new job has been submitted,

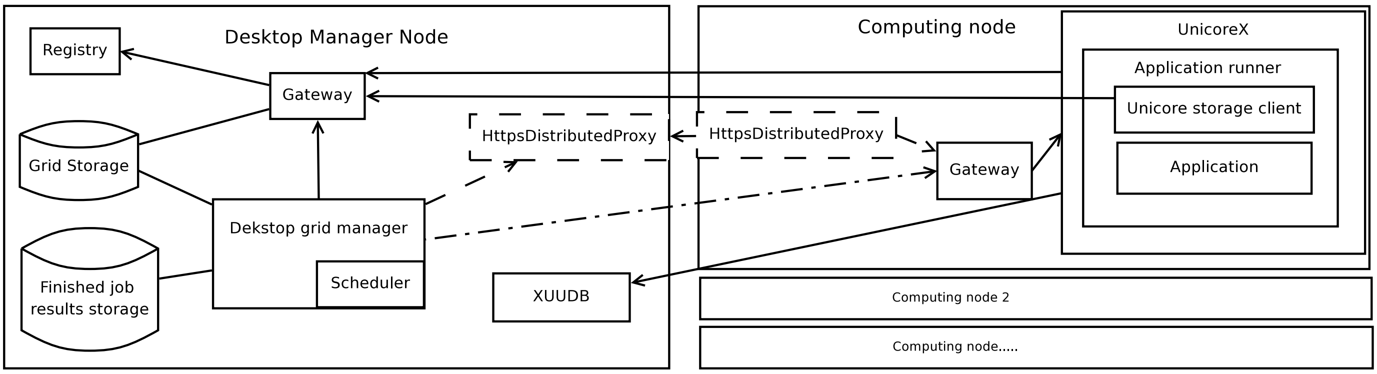


Fig. 2. Architecture of desktop grid

- node has finished computations,
- manager task has finished,

the Desktop Manager tries to run a new task on free computing elements. It runs a scheduler module which is responsible for matching tasks with the computing elements. The actual running and fetching of a task to a private computer is done by Node Manager which is part of Desktop Manager.

*4) Fetching Sub-Tasks Results:* When a job finishes computations, which means that all data is sent to a grid storage (see III-A1),the node monitoring thread receives information that the job has finished (by executing Unicore check job state call). Then, it copies all data from desktop grid storage to the finished job storage, and tries to perform the rescheduling. Or, possibly, if all tasks for the job have finished, it tries to queue a new manager thread for merging the results.

*5) Monitoring State of Nodes:* The system monitors two indicators for node activity:

- node down or up,
- state of job on the node

Because the registry can have out-of-date information about nodes, we have to monitor its status by our own. For this purpose, we call the Unicore check node time call, which is a common method of testing if the whole node Unicore infrastructure is running.

*6) Use of Desktop Grid Manager:* Desktop Grid Manager is built of two parts—core manger described above and the User interface which can be easily modified and adopted to the user's needs. Different possible settings of usage of the desktop grid are presented in Fig. 3, Fig. 4, Fig. 5 and Fig. 6.

In Fig. 3 Desktop Grid Manager plays a role of target system. Additionally, because the desktop grid uses the same middleware as Unicore grid, we can utilise free time of Unicore grid computing nodes.

In Fig. 4 there is presented a desktop grid as a standalone system. This would be achieved by changing authorisation database for Desktop Grid Manager from Unicore grid authorisation database to local one.

Standalone settings could be slightly modified by allowing clients to be computing units at the same time. Such settings are presented in Fig. 5.

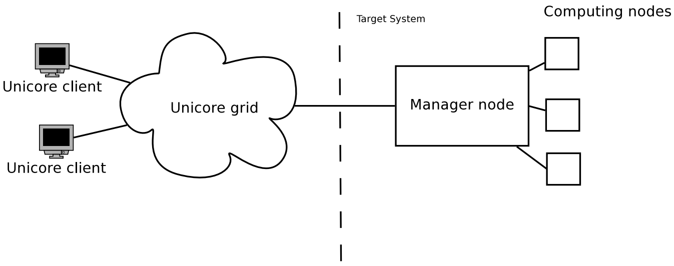


Fig. 3. Desktop grid manager server as target system for Unicore

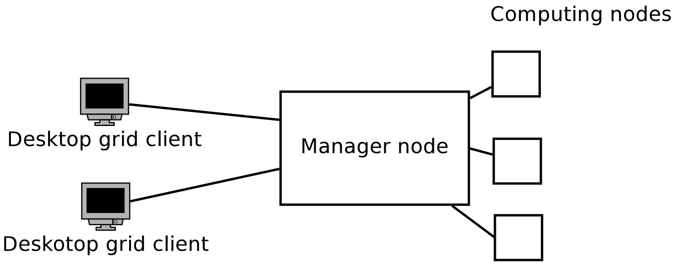


Fig. 4. Desktop grid manager as standalone server

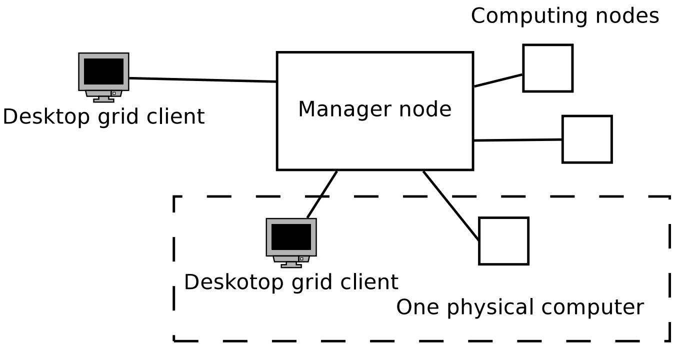


Fig. 5. Desktop grid manager as standalone cooperative system

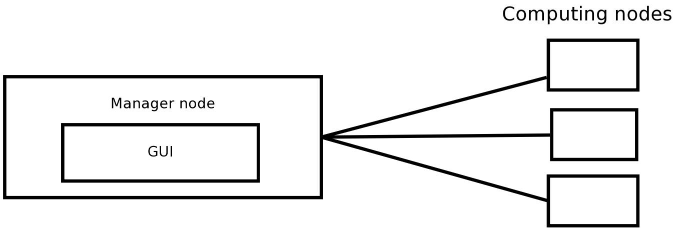


Fig. 6. Desktop grid system—development setting

Finally, Fig. 6 presents a currently implemented development setting.

With respect to this setting we created a simple GUI for Desktop Grid Manager. This solution makes monitoring of jobs execution really easy and, moreover, it simplifies debugging.

*7) Scheduler:* Scheduler used in our Desktop Manager is a random scheduler with added backup policy. It tries to assign tasks with lower number of running instances first. If tasks have the same number of running instances, then the one belonging to job that was assigned first is run.

### *D. HttpsDistributedProxy*

As it is presented in Fig. 2, the system could contain distributed https proxy. This part has already been implemented but it hasn't been incorporated into the desktop grid yet. Because the computing elements could be put behind firewall or NAT, we introduce https distributed proxy. This proxy is built of two parts:

- server part—where https client connects and asks for page
- computing node part—the part that opens a connection to a server part and waits for data coming from server, that should be tunnelled to a site.

Because https protocol does not allow looking into packets by the proxy, the tunnelling is the only available option. Because the whole communication is started by a computing node part, the computing node may be located behind NAT gateway and be not visible from Internet.

### *E. Security*

The security in the Desktop Grid is based on X.509 certificates. Desktop Server has one certificate, and every computing elements group should have their own one, too. Because certificates should be generated when the owner of computers gets a software package, and we cannot guarantee that few instances of one package will not work at the same time, we introduce a computing elements group—i.e. different computers working with the same certificate set.

Additionally, every sub-task is given its own certificate that will allow getting and putting on the desktop grid storage only such files which belong to it.

Currently, our desktop grid works with one set of certificates—all parts use the same certificate.

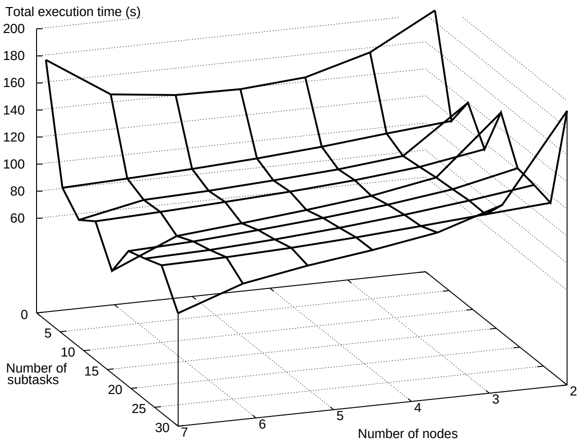


Fig. 7. Efficiency of presented system

## IV. Efficiency Tests

Desktop grid was tested on naive implementation of concurrent Mandelbrot set computing algorithm. For purposes of tests we used jobs that took 236 seconds on a single computer (run locally).

In Fig. 7 there are presented results of efficiency tests for the created system. On one axis there is presented the number of working nodes (manager node is not counted), on another one the number of sub-tasks which the job was divided to. What seems unclear here is a fact that the minimum time of execution has been achieved for a number of tasks larger than the number of nodes(not slightly larger but double or triple). This is caused by pure balancing of tasks in naive concurrent version of algorithm.

In Fig. 8 there is presented the speedup of computations as a function of number of nodes. Speedup is a time taken by computations on one machine, divided by minimum total execution time for specified number of nodes. Below 5 nodes this function is of linear nature, however worse than optimum line $y = x$. Above 5 nodes, the constant cost gains in importance, thus making the difference in speedup between 6 and 7 nodes much smaller than that between 3 and 4 nodes. These results show that our system is quite effective, although it should be optimised.

## V. Conclusions

Our work shows that Unicore 6 could be used as a basic middleware for desktop grid. It is easy to be configured and it needs only small efforts to develop Desktop Grid Manager.

## VI. Future Work

In the future we plan to:

- incorporate https distributed proxy to desktop grid,
- incorporate architecture for managing certificates from project Chemomentum [8], and add Uvos in a further step,
- complete detaching the application run by nodes from disks—by using Java policy,

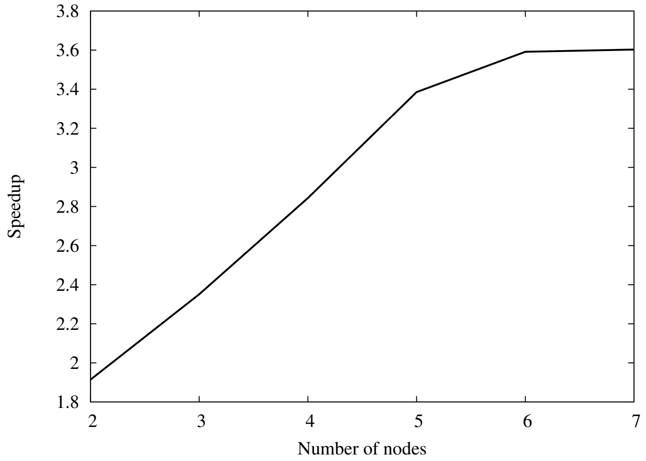


Fig. 8. Scalability of system—speed up

- attaching Java code to be executed, as an part of a task, to the task submission.

## Acknowledgment

Work supported by the joint project of ICM UW and Telekomunikacja Polska S.A. 22/06/6727/K/2006/YCZ268 Grid System Monitoring and Control Tools under grant FSO 023/2006.

## References

[1] *Installation and Configuration of UNICORE 6* http://www.unicore.eu/documentation/manuals/unicore6/files/Installation_UNICORE6.pdf

[2] David P. Anderson, *BOINC: A System for Public-Resource Computing and Storage*, 5th IEEE/ACM International Workshop on Grid Computing. November 8, 2004, Pittsburgh, USA.

[3] Douglas Thain, Todd Tannenbaum, and Miron Livny, *Distributed Computing in Practice: The Condor Experience*, Concurrency and Computation: Practice and Experience, Vol. 17, No. 2–4, pages 323–356, February–April, 2005.

[4] Konstantinos Georgakopoulos, Konstantinos Margaritis, *Integrating Condor Desktop Clusters with Grid*, Distributed and Parallel Systems In focus: Desktop Grid Computing, September 2008.

[5] Zoltán Farkas, Péter Kacsuk, Manuel Rubio,*Utilizing EGEE for Desktop Grids*, Distributed and Parallel Systems In focus: Desktop Grid Computing, September 2008.

[6] Ian Kelley, Ian Taylor, *Bridging the Data Management Gap Between Service and Desktop Grids*, Distributed and Parallel Systems In focus: Desktop Grid Computing, September 2008.

[7] A. Faroughi, R. Faroughi, P. Wieder, W. Ziegler, *Attributes and VOs: Extending the UNICORE authorisation capabilities*, Proceedings of 3rd UNICORE Summit 2007 in conjunction with EuroPar 2007, Rennes, France, LNCS 4854, pages 121–130.

[8] B. Schuller, B. Demuth, H. Mix, K. Rasch, M. Romberg, S. Sild, U. Maran, P. Bala, E. del Grosso, M. Casalegno, N. Piclin, M. Pintore, W. Sudholt, K. Baldrige, *Chemomentum—UNICORE 6 based infrastructure for complex applications in science and technology*, Proceedings of 3rd UNICORE Summit 2007 in conjunction with EuroPar 2007, Rennes, France, LNCS 4854, pages 82–93.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Load balancing in the SOAJA Web Service Platform

Richard Olejnik*, Iyad Alshabani*, Bernard Toursel*, Eryk Laskowski†, Marek Tudruj†‡
*Computer Science Laboratory of Lille (UMR CNRS 8022). University of Sciences and Technologies of Lille, France.
†Institute of Computer Science Polish Academy of Sciences, Warsaw, Poland
‡Polish-Japanese Institute of Information Technology, Warsaw, Poland
{Richard.Olejnik, Iyad.Alshabani, Bernard.Toursel}@lifl.fr
{laskowsk, tudruj}@ipipan.waw.pl

***Abstract*—The aim of the Service Oriented Adaptative Java Applications (SOAJA) project is to develop a service-oriented infrastructure, which enables efficient running of Java applications in complex, networked computing Grid environments. The SOAJA environment provides components and services for static and dynamic load balancing based on Java object observation. SOAJA can be used to design large scale computing tasks to be executed based on idle time of processor nodes. Java distributed applications consist of parallel objects which SOAJA allocates to Grid nodes at runtime. In this paper, we present mechanisms and algorithms for automatic placement and adaptation of application objects, in response to evolution of resource availability. These mechanisms enable to control the granularity data processing and distribution of the application on the Grid platform.**

***Index Terms*—Service Oriented Applications, Adaptative Applications, Load balancing, Grid Computing Distributed Computing**

## I. Introduction

LOAD balancing is one of important procedures applied to heuristically optimize execution time of parallel programs. A general classification and an overview of load balancing methods are presented in [3], [4]. The paper deals with an asynchronous approach to load balancing, in which the load balancing activities are performed in parallel with computations. Load balancing can be further divided into static load balancing where an computational load is partitioned among executive units by an algorithm executed before program execution and dynamic load balancing, where the load decomposition is adaptively changed during computations, following the system resources availability. This paper is concerned equally with static and dynamic load balancing. In Java-based computing on Grid static load balancing has received relatively small attention. In this paper we present a two phase approach to load balancing of Java programs execution in Grid. The first phase consists of a static load balancing, which determines an initial deployment of application Java objects over the network of Java Virtual Machines. This static load balancing algorithm scenario includes execution of the application for a representative set of data to be able to detect some static properties concerning computational and communication aspects and to be able to use this properties for an initial deployment of program elements before execution. This phase of load balancing is based on tracing of the load of virtual machines and method invocations. The second phase of the load balancing process is dynamically organized during program execution. It is based on three basic operations: JVM load observation, detection of the load imbalance and load migration if the imbalance exists. A dynamic agent approach is used to implement these operations. Some metrics to detect and measure the load imbalance of processors have been proposed in the paper. They differ from standard measures known in the literature [2].

This paper describes SOAJA overall architecture and then deals with its internal concepts. The rest of the paper is composed of 5 parts. In the first part the general assumptions for the SOAJA framework are presented. Part 2 explains the use of web services in SOAJA. Part 3 discusses the relations between web services and functions of DG-ADAJ. Part 4 describes the load imbalance detection mechanisms in SOAJA. Part 5 describes the load imbalance correction mechanisms.

## II. SOAJA and Grid

The SOAJA (Service Oriented Adaptative Java Applications) infrastructure provides components and services enabling a platform-independent access, sharing and application of potentially distributed complex data mining workflows [7] and resources, including database and information systems and hardware resources. It supports resource discovery and will supply context-aware recommendations for the dynamic composition of data mining operations and workflows. The underlying agent-based layer of the SOAJA infrastructure will provide means to orchestrate very large, heterogeneous and dynamic hardware and software resources across multiple platforms. The SOAJA is deployed in a grid infrastructure with a JVM on each processor node. The main services of this framework are observation, measuring of the JVM load, measuring of the physical processor load, load balancing service, and data parallel services. The SOAJA environment is the extension of the ADAJ environment to make it scalable on the Grid and to support service oriented architecture [5].

ADAJ is a programming and execution environment for parallel and distributed applications, which facilitates the design and optimizes performance. ADAJ is a Java environment based on JavaParty [14], which optimizes the RMI protocol. The JavaParty allows execution of distributed Java applications on workstations connected via a network. JavaParty has introduced the concept of remote objects that can be distributed in a transparent manner. It compensates for the drawbacks of the RMI protocol because it conceals the addressing and

978-83-60810-14-9/08/$25.00 © 2008 IEEE

communication mechanisms. Indeed, it is sufficient to annotate Java classes with the word “remote” that will give access to remote objects from any JavaParty environment without publishing them explicitly in service names space as RMI.

Unfortunately, designing distributed programs and optimizing their performance does not remain as simple. Indeed, a programmer must consider construction of its application in the most effective way, while taking account of heterogeneity of the executive environment. This is what the role of ADAJ is. Principles of ADAJ:

- Simplifying the programmer’s work by hiding problems related to the management of parallelism,
- Facilitating the development of applications and allowing their automatic or quasi-automatic deployment in heterogeneous environments,
- Ensuring effective implementation of parallelism, by mechanisms of inter- and intra-applications load balancing.

ADAJ includes four major features:

- A library containing the necessary tools to facilitate parallel programming,
- An observation system which scans the environment during its execution and retrieves the information necessary to optimize the program,
- A system that calculates the load of JVM and physical processors using the information gathered by the observation system,
- A system that allows correcting load imbalance by objects migration based on information from the the observation and the calculation of loads.

DG-ADAJ is the ADAJ environment implemented for Desktop-Grid. Initially, DG-ADAJ was conceived as an extension of the ADAJ system (see [10]), built for cluster computing. It has been re-engineered to extend its for larger scale distributed computing and to introduce some special security mechanisms, which provide reliable application execution [12].

The Figure 1 illustrates the main web services that belong to SOAJA environment.

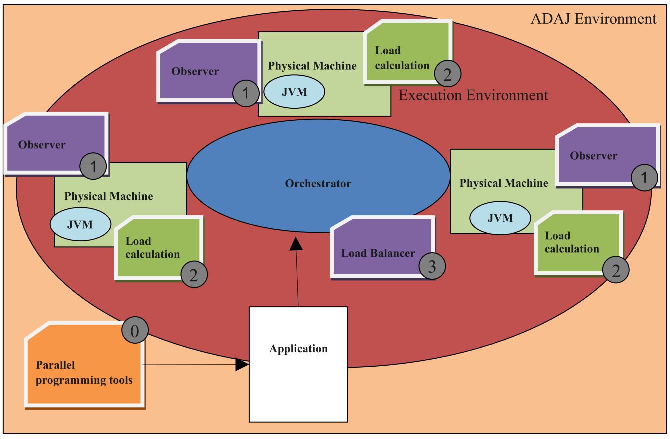


Fig. 1. SOAJA Environment

SOAJA is based on WSRF [9] and allows instantiation of statefull services for applications. The statefullness of the services is exploited in relation with application clients that will allow asynchronous communication with software components instantiated for the client on different platforms.

Various middleware components were considered to enable SOA services. In close relation with these components, specific technologies were defined and implemented in order to achieve interoperability. The Enterprise Service Bus (ESB) is a middleware technology providing the necessary characteristics in order to support SOA. In a typical ESB architecture, the ESB layer itself is deployed over the existing infrastructure. Based on this infrastructure, the ESB layer offers the necessary support for transport interconnections while it exposes the existing subsystems through a specific set of adapters.

With the help of the ESB, services are exposed in a uniform manner, based on open standards, so that any client, who is able to consume web services over a generic or specific transport, is able to access them. For example, the SOAJA has been used to implement data mining application in the WODKA project [11].

## III. ADAJ Web Services

SOAJA is a platform developed on top of DG-ADAJ, by adding the web services layer and gaining SOA-specific properties. With computing grid as a central target, the SOAJA platform provides a uniform and transparent interface to the infrastructure of the DG-ADAJ platform. The SOAJA platform, through its orchestration layer, implements an alternative to facilitate execution in the DG-ADAJ environment [1], [13]. With DG-ADAJ executions controlled via an ESB, the orchestration layer is able to offer both the support for execution of complex compositions, as well as elementary execution of the underlying DG-ADAJ environment.

The solution is hiding implementation details of different Grid environments behind various web service standards and technologies, offering the necessary support for integration, interoperability, and reliable messaging. As a side effect, by employing the orchestration layer, we enable a programming-in-the-large paradigm necessary to assure the development and support of long living, asynchronous processes.

The SOAJA platform is going to help efficient execution of heterogeneous applications enabled by DG-ADAJ by offering basic support for workflow deployment and enactment. The DG-ADAJ environment could thus be freed from some of the placement, distribution and execution tasks, by moving significant parts of these to the higher layer of the ESB and the enactment environment. With SOAJA, the ability to design component-based and service oriented applications, developed over the DG-ADAJ platform is extended with the help of web services open standards, by offering the possibility to access both local and remote components, eventually deployed on different DG-ADAJ environments.

We have found no tool to transform packages containing many classes, directly into a web service package. We had to go through the implementation of interfaces to identify methods that might be invoked by other web services.

Let us now describe the internal mechanism of DG-ADAJ.

## IV. Execution optimization in SOAJA

Distribution of the application components (objects) among active Grid nodes should guarantee a possibly high efficiency of the overall application execution. Thus, the following two aspects of execution optimization have been taken into account in the SOAJA environment:

- initial objects deployment,
- dynamic load balancing.

### *A. Initial object deployment optimization*

An optimization feature of SOAJA is an initial application objects deployment on JVMs, which results in a shorter execution time. The initial placement of application objects to JVMs is the service of the orchestrator shown in Fig. 1. The initial object deployment optimization algorithm follows the pattern of static parallelization method in multithreaded Java program. It defines decomposition of Java code into parallel threads distributed on a set of JVMs, so as to reduce program execution time [13]. The control decisions are taken to determine which of the classes should be distributed and what the mapping of objects and data components (fragments) should be to JVM nodes, to reduce direct inter-object communication and to balance loads of the JVMs. When applied to an application run under DG-ADAJ control, it will determine an initial distribution of its objects among Java Virtual Machines (JVMs) assigned to Grid active nodes, thus leading to a reduction of the total execution time.

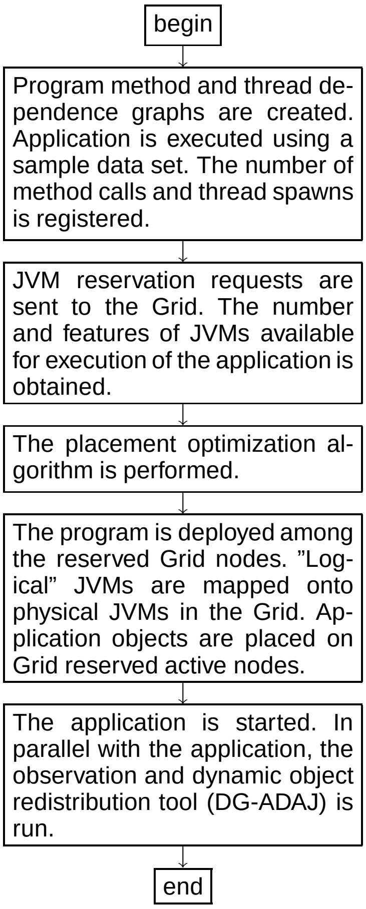
The proposed optimization algorithm employs an image of application program, based on an analysis of the byte code generated by Java compiler. This analysis identifies control dependencies between byte code instructions. They are represented in adequate MDG (Method Dependence Graph) and MCG (Method Call Graph) graphs of the program [6], [13]. The number of mutual method calls and the number of thread spawns during program execution for representative input data are measured using observation mechanisms described later in the paper. Program behavior, including all created objects in each class, all called methods, as well as all spawned threads, are registered in trace files.

The flow of actions during application execution is shown in the diagram in Fig. 2. The first three blocks in the diagram determine an initial optimized placement of application objects and perform the respective objects distribution over Grid on JVMs nodes. This part of the algorithm starts with execution of the application using some representative sample data. For that, the number of available JVMs nodes on the Grid must be known. The number of method calls and spawned threads that have appeared during execution is recorded. Next, the program method and thread dependence graphs are annotated with the recorded data.

At the beginning of the object deployment optimization algorithm we treat all objects as remote objects (respectively all classes are distributed classes). Based on the recorded control data, the algorithm decides which classes should remain distributed and how the involved objects should be placed on a set of JVMs assigned to active nodes on the Grid. We use the heuristics based on the following principles: the strong locality of method calls has to be preserved inside each parallel thread and the number of inter-thread calls, which cross the boundaries of JVMs has to be reduced. To fulfill such object requirements we designate all calls inside a single thread to the same JVM and optimize distribution of threads across available sets of JVMs by applying load balancing methods.

The algorithm consists of two phases (see [6] for details). In the first phase the MCG graph is traversed in the DFS (Deep-First-Search) manner to agglomerate method calls executed in single thread. In this step, the algorithm finds the MCG subgraphs, which are constructed of vertices connected by edges connecting calls inside threads. We assume that each subgraph is executed on a dedicated JVM. Subgraphs are built at the level of single objects. In case when an object belongs to different subgraphs, the new subgraph is constructed and a unique JVM number is assigned to it. We expect that, in most cases, at the end of this phase, the number of found subgraphs is far bigger than the number of available JVMs in the system.

In the second phase of the algorithm, we clusterize the



Program method and thread dependence graphs are created. Application is executed using a sample data set. The number of method calls and thread spawns is registered.

JVM reservation requests are sent to the Grid. The number and features of JVMs available for execution of the application is obtained.

The placement optimization algorithm is performed.

The program is deployed among the reserved Grid nodes. ”Logical” JVMs are mapped onto physical JVMs in the Grid. Application objects are placed on Grid reserved active nodes.

The application is started. In parallel with the application, the observation and dynamic object redistribution tool (DG-ADAJ) is run.

Fig. 2. The control flow of an application execution.

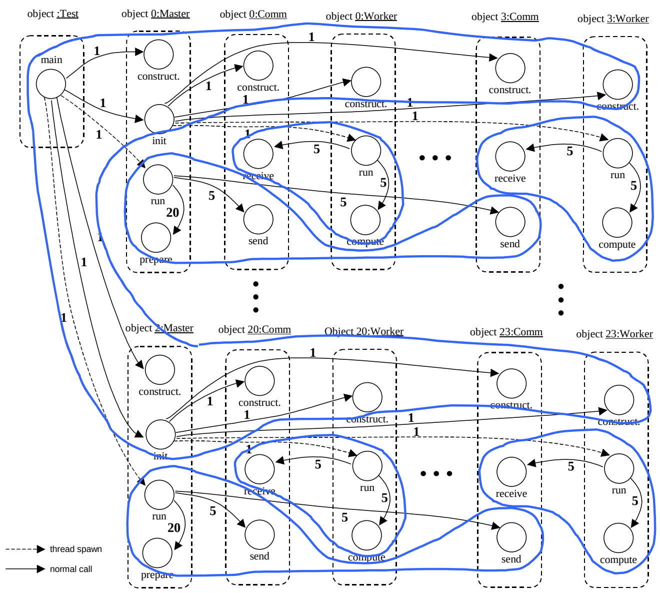


Fig. 3. A MCG for a Java program with threads' sub-graphs shown.

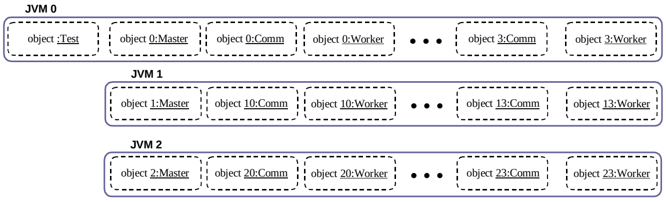


Fig. 4. Final distribution of application objects across JVMs.

subgraphs obtained in the previous phase until the number of clusters is equal to the number of JVMs. The general outline of this phase is similar to Sarkar's [15] edge-zeroing clustering heuristics. At each clustering step, the algorithm finds the subgraphs, which are connected by edges with the biggest weight value and which connect nodes placed on different JVMs. All nodes of those subgraphs are assigned to the same JVM while the total number of remote calls decreases. The algorithm stops when the number of clusters is equal to the assumed number of JVMs.

An exemplary result of the first phase of the algorithm is shown in Fig. 3. A MCG has been partitioned into sub-graphs and objects, which belong to more than one sub-graph, are assigned to unique JVM. During clustering phase, objects that frequently invoke methods in different objects are moved to the called object's JVM. Final distribution of program objects across JVMs is shown in Fig. 4.

The first phase of the algorithm makes that method calls inside programmer-declared threads are local to the JVM. This allows exploiting thread level parallelism without introducing large inter-JVM communication overheads. The gain of the second step of the heuristics comes from reduction of RMI calls to remote objects.

## *B. Load balancing*

The workstations used in the network are heterogeneous, but they have different and variable computing capabilities over time. The load imbalance occurs when the differences in workload between the workstations become too big. An application execution may not be optimal because some workstations have too much work and the others have not enough. Let's consider that the network works correctly and that the DG-ADAJ environment is running. We distinguish two main steps in load balancing: detection of imbalance and its correction, if necessary. The first step uses measurement tools to know the functional state of workstations. The second consists in migrating of the load from overloaded workstations to underloaded workstations in order to balance the workload.

The observation mechanism of applications in the DG-ADAJ environment aims at providing knowledge of the applications behavior during their execution. This knowledge is gathered by observing activity of constituent objects.

There are two types of objects in DG-ADAJ (Fig. 5):

- **global objects:** These are global objects that can be created remotely in any JVM. They are remote accessible. There is only one copy of a global object in all environment. The global objects are also migratable, i.e. they can be moved from one JVM to another.
- **local objects:** These objects are traditional Java objects. They can be used in only one JVM at the place where they reside. If another JVM needs such object, it will create of a new copy of the object concerned. Obviously, local objects cannot be migrated.

We decide to observe only global objects as the observation and migration of all objects would generate a considerable work overhead compared to the profit brought by load balancing.

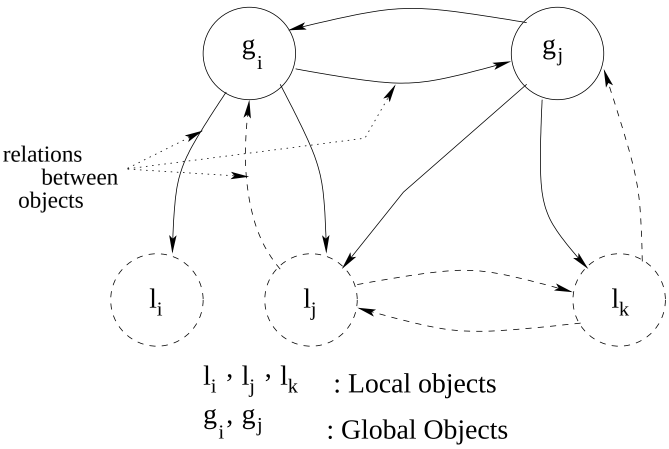

$g_i$
$g_j$
relations between objects
$l_i$
$l_j$
$l_k$
$l_i$ , $l_j$ , $l_k$ : Local objects
$g_i$, $g_j$ : Global Objects

Fig. 5. Relations between Objects

## V. Detection of load imbalance

In this first phase, the purpose is to obtain knowledge about of the functional state of workstations composing the cluster. As the environment is heterogeneous, it is necessary to know not only the load of workstations but also their working capabilities. Such capability is directly related to the workstation computing power, so we have to estimate it. This measurement needs to be made only once when the workstation joins the system. We call it *the calibration* in this paper.

We then need the workstations workload measurement at a given time. For this purpose, we observe this part of the CPU time that is used by DG-ADAJ applications. However, such measurement is not normalized and cannot serve for comparisons between different workstations. It is thus necessary to balance it against different computing power of the workstations. After a series of measures, we compare the values found on different machines and we determine if there is a load imbalance.

The heterogeneity disallows us to compare measurements taken on workstations whose computing powers are different. Before we are able to compare the workstations load, we have to normalize the measurements. The normalization of the workloads is made by means of the power indications. After experiments to determine the workstation power, we found that the formula, which allows us to compare the workstations load is the product of the power index and the CPU time use rate by a thread, which gives the availability index of a CPU:

$$Ind_{availability} = Ind_{power} * \%Time_{CPU}$$

At this point, we have availability indices of all workstations. By comparing these indices, we will be able to detect the load imbalance. An imbalance is characterized by too big dispersion of the availability indices of workstations composing the network. It is difficult to fix a threshold not to be overpassed to characterize a load imbalance. But we can define an interval in which the dispersion of the availability indices remains acceptable at a given moment. We are going to be interested in the gap between the minimal and maximal availability indices found during a series of measures. If the distance between these values is too big, we conclude that there is a load imbalance, which can to be considered using the following condition:

$$Stability = (max(Ind_{availability}) \leq \alpha * min(Ind_{availability}))$$

If this inequality is verified, the availability indices are close enough not to present a too big imbalance. Otherwise, the range of the values is too big and an imbalance of workload between workstations is detected. The central point of this inequality is the value taken by the coefficient $\alpha$. We tried to clarify it first by using statistical tools then by using an experimental way. We cannot finally give a unique value for this coefficient. Nevertheless, we can restrict its value to the interval $[1.5 \ldots 2.5]$. These experimental values give good results because they are neither too restrictive nor too tolerant for the load imbalance.

The tolerance for the load imbalance among all workstations depends directly on the value of the coefficient $\alpha$. The smaller this coefficient is, the more we are demanding as to the load balance between workstations. Indeed, a low coefficient allows only a small difference between loads of workstations. This can be seen as a guarantee of the quality of the balancing results. However, a low tolerance of the imbalance leads to much more frequent detection of the imbalance. The consequence is to activate the mechanism of load balancing more frequently and therefore make it more expensive in time. Not only the migrated objects can not do their computing, but they can also block the execution of others objects, which are awaiting results. In addition, the coefficient $\alpha$ is to be chosen according to the number of machines in the network. When this number is big, the coefficient must be increased and vise versa.

## VI. Correction of load imbalance

### A. Classification of workstations

In this phase, we are in the state in which a load imbalance has been detected. To correct this imbalance, we have to classify the workstations into three categories according to their availability index:

- Overloaded workstations: availability indices are low,
- Normally loaded workstations: availability indices are medium,
- Underloaded workstations: availability indices are high.

The purpose of the load balancing is to transform the workstations categories: overloaded and underloaded into the category normally loaded. To do it, we need to migrate the workstations load from the first category to the third one.

We use the K-Means algorithm [8] to build the categories of workstations based on the computed availability indices. The K-Means algorithm allows to classify a distribution of $n$ values into $k$ categories by choosing $k$ centers for categories. We want to classify workstations into the categories: overloaded, normally loaded and underloaded. For this, we use this algorithm by taking the computed availability indices and $k = 3$, to obtain 3 categories finally.

The three centers that we choose are the minimum, average and maximum availability indices. The average index is simply the average of indices measured during the last series of measures over the whole network. By comparing the distances of workstations availability indices from the three centers, the three categories of workstations will be identified. The center represented by the minimum index builds the category of overloaded workstations, the center using the average one builds the category of normally loaded workstations and finally the center based on the maximum index is used to build the category of underloaded workstations. The important thing is therefore to have the overloaded and underloaded categories in order to be able to move the load from the overloaded workstations to the underloaded workstations.

### B. Choice of candidates for migration

To correct load imbalance, we have to migrate the load from overloaded workstations to underloaded workstations. Firstly, we must identify the load that we want to migrate. The loads are represented by the activities of the objects which are running on the JVMs. We need to select an object on each JVM in the overloaded nodes. Let's see how to choose such an object, so that its migration changes the load balance in the network.

The migrated entity is necessarily a global object because it is not intrinsically linked to the JVM on which it currently runs. Among the global objects in a JVM, some of them have more suitable characteristics to be migrated. These characteristics are related with the other computer objects and the load quantity carried out. Two relations are involved:

- the attraction of a global object to the JVM,
- the weight of the global object.

The attraction of a global object to a JVM is expressed in terms of communication links, which it shares with other global objects inside the same JVM on the computing node. A strong attraction involves frequent communication, which will be realized as remote communication after object migration. This communication will then have a higher cost than the current cost. A small attraction will permit to leave a global object the current computing node and to run it on another one, without introducing significant amount of additional remote communication. So, the less the object is attracted by the current JVM, the more interesting it is to be selected as a migration candidate.

The computational weight of migrated global object gives the quantity of load to be removed from the current machine. An object, whose quantity of work is big i.e. shows a continuous activity, should not be migrated. In addition, by migrating a too big quantity of work (load), we could reverse the role of the involved machines (the source will become target and vice-versa). In contrast, the migration of an object with small quantity of a work does not bring significant load variation improvements. Furthermore, the migration cost will not be compensated with the new generated load distribution. In conclusion, the decision should be to move an object whose quantity of work is neither too big, nor too small. Thus, the smaller the distance is to the average object loads, the more the object is interesting for migration.

The observation of the objects activity is done by counting the activation methods. These activations can be done by global or local objects. The observation of a global object (only these objects are observable), includes:

- observation of object invocations to each global object, including him: OGI (OutputGlobalInvocation),
- the observation of object invocations to all local objects: OLI (OutputLocalInvocation),
- the observation of others objects invocations to the considered object: II (InputInvocation).

The attraction of the global object $obj$ to the actual JVM:

$$attr(obj) = \sum_{o \in JVM} (OGI(obj, o) + OGI(o, obj))$$

Distance compared to the average quantity of work of the $obj$:

$$dist_{m_{WP}}(obj) = |WP_{obj} - m_{WP}|$$

where $m_{WP} = \frac{\sum_{o \in JVM} WP_{obj}}{n}$ ($n$ is the number of global objects on the JVM) and
$WP_{obj} = OGI(obj, obj) + II(obj) + OLI(obj)$.

These formulas allow to compute the attraction of an object to the local JVM in order to compare it with the attractions of other objects of this JVM. The comparison formulas are:

- The percentage of the global attraction:

$$\%attr(obj) = \frac{attr(obj)}{\sum_{o \in JVM} attr(o)}$$

- The percentage of distance, compared to the average quantity of work of the object:

$$\%dist_{m_{WP}}(obj) = \frac{dist_{m_{WP}}(obj)}{\sum_{o \in JVM} dist_{m_{WP}}(o)}$$

Finally, we compute the weighted sum of these relations in order to determine the most interesting object to migrate:

$$Classification(obj) = \alpha_{attr} * \%attr(obj) + (1 - \alpha_{attr}) * \%dist_{m_{WP}}(obj)$$

$\alpha_{attr}$ is a real between 0 and 1. Its choice remains experimental. Let us notice however that the bigger $\alpha_{attr}$ is, the bigger is the added weight to the object attraction.

### C. Selection of the target for migration

The migration of global objects reduces the load of a JVM running on an overloaded workstation and therefore the global load cost of this workstation. The question now is: where migrate these objects? Naturally, the potential destinations are one or more underloaded workstations. However, the choice of one of these computing nodes can be more or less convenient from the point of view of work and communication quantity of the object to migrate. For an object selected for migration, we must find the best target according to these criteria.

The first criterion to quality as a target is the attraction of the selected object to this workstation. We say that the attraction of an object – candidate for migration, is big when it communicates a lot with the global objects in the target JVM. A relationship of attraction of the global object $obj$ to $JVM_i$ is defined as follows:

$$attrext_i = \sum_{objext \in JVM_i} (OGI(objext, obj) + OGI(obj, objext))$$

The more the object is externally attracted by an underloaded machine, the more it is interesting that this machine is chosen as a migration destination for it. This criterion should not be the only one during selection of target for migration. In fact, it is possible that two underloaded workstations have the same amount of communication with the candidate for migration. In this case, the second criterion will be the workstations' availability indices. We naturally prefer the one whose availability index is the highest, because it is actually the least loaded.

To complete discussion of the criteria for choosing a target for migration, we should take into account the number of *(waiting)* threads in the JVM of the potential targets. We consider them, however, as potential load, which must be taken under consideration with the related load currently done on the machine.

These three points are obvious constraints to be met by the target machine:

- the external attraction of the object is maximal to the target machine,
- the quantity of work on the target machine is minimal,
- the number of waiting Java threads is minimal.

These three conditions are gathered in a formula in order to designate the underloaded workstation, which is the most favorable for the selected object migration. Firstly, we have to normalize all the values related in the interval $[0\ldots1]$. We then obtain:

$$\%attrext_i = \frac{attrext_i}{\sum_j attrext_j}$$

$$\%Ind^*_{availability_i} = \frac{Ind^*_{availability_i}}{\sum_j Ind^*_{availability_j}}$$

where

$$Ind^*_{availability} = Ind_{availability} - Ind_{availability} * \frac{NbThread_{wait}}{NbThread_{total}}$$

The availability index was corrected by the potential work of the workstation, represented by the waiting threads. We have not considered this index during the classification of workstations into three categories, because we want to classify workstations according to their measured quantity of work and not a potential one. Indeed, threads waiting must be seen as supplementary work about which we know nothing. They may start executing in one second quite well as in one hour. Their consideration is thus justified only when we want to examine our measurements in perspective as it is the case here. We thus chose to decrease the raw indication of availability by means of the relationship between the number of waiting threads and the total number of threads staying in the machine.

The aggregation function should account for the two described components by giving them different weights. The balanced sum of them is:

$$Quality_i = \alpha_q * \%attrext_i + \beta_q * \%Ind^*_{availability_i}$$

with $\alpha_q$ and $\beta_q \in [0\ldots1]$

For an object which is a candidate for migration, this formula is applied to all JVM potential node targets. The workstation which maximizes this sum will be chosen as new location for the object. The choice of coefficients $\alpha_q$ and $\beta_q$ is experimental but the sum of the two must be equal to 1 (it was therefore $\alpha_q = 1 - \beta_q$). The weight of one or the other value can be increased by changing these coefficients. The migration of the object allows to eliminate communication on the network and to reduce considerably the waiting time for replies. For that purpose, the coefficient $\alpha_q$ must be the most important to promote the machines for which the attraction of the object is maximal. For example, we can use the coefficients $\alpha_q = 0.6$ and $\beta_q = 0.4$.

## VII. Conclusions

This paper has presented the mechanisms and algorithms, which ensure automatic support for Java distributed programs execution on Grids. It includes initial placement of the application objects to the current used host system configuration and further dynamic program execution adaptation in response to the computing system evolution and to modifications of the resource availability. The proposed program execution optimization mechanisms provide control to adjust the initial granularity of the application program parallelization and the dynamic re-distribution of the application on the Grid platform. The SOAJA infrastructure with its observation mechanisms provides components and services enabling static and dynamic load balancing using idle CPU time of the nodes of a Grid. The system is currently under implementation based inside the GRID 5000 project.

## References

[1] I.Alshabani, R. Olejnik and B. Toursel. *Parallel Tools for a Distributed Component Framework.* 1st International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA04). Damascus, Syria, April 2004.

[2] J. Cao et al., *Grid load balancing using intelligent agents*, Future Generation Computer Systems, 21 (2005), pp. 135–149.

[3] K. Devine et al., *New challenges in dynamic load balancing*, Applied Numerical Mathematics, 52 [2005], pp. 133–152, Elsevier.

[4] R. Diekmann, B. Monien, R. Preis, *Load Balancing Strategies for Distributed Memory Machines*, Karsch/Monien/Satz (ed.): "Multi-Scale Phenomena and their Simulation" World Scientific, pp. 255-266, 1997.

[5] T. Erl, *Service-oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River: Prentice Hall PTR, 2005. ISBN 0-13-185858-0.

[6] V. Felea, E. Laskowski, B. Toursel, M. Tudruj, *Optimizing Object Oriented Programs Based on the Byte Code-Defined Data Dependence Graphs*, Procs. of Concurrent Information Processing and Computing (CIPC NATO ARW), Sinaia, Romania, pp. 34–46, 2003.

[7] V. Fiolet, G. Lefait, R. Olejnik, B. Toursel, *Optimal Grid Exploitation Algorithms for Data Mining*, In Proc. of ISPDC 2006, IEEE Computer Society, July 2006, pp. 246–252.

[8] J.A. Hartigan et M.A. Wong, *A K-Means clustering algorithm*, Applied statistics, Vol. 28, pp. 100-108, 1979.

[9] OASIS Web Services Resource Framework (WSRF) http://www.oasis-open.org/committees/wsrf/

[10] R. Olejnik, A. Bouchi, B. Toursel. *An Object Observation for a Java Adaptative Distributed Application Platform.* Intl. Conference on Parallel Computing in Electrical Engineering PARELEC 2002, pp. 171–176, Warsaw, Poland, September 2002.

[11] R. Olejnik, F. Fortis, B. Toursel, *Webservices Oriented Datamining in Knowledge Architecture*, Accepted to publication in Future Generation Computer System (FGCS)—The International Journal of Grid Computing: Theory, Methods and Applications

[12] R. Olejnik, B. Toursel, M. Ganzha, M. Paprzycki, *Combining Software Agents and Grid Middleware*, Advanced in Grid and Pervasive Computing, C. Cerin and K.-C Li Editors, LNCS 4459, pp. 678–685, Springer Verlag, Berlin, Heidelberg, 2007.

[13] Olejnik R., Toursel B., Tudruj M., Laskowski E., *Byte-code scheduling of Java programs with branches for desktop grid*, Future Generation Computer Systems, Vol. 23, Issue 8, November 2007, pp. 977–982, ©Elsevier Science.

[14] M. Philippsen, M. Zenger. *JavaParty – Transparent Remote Objects in Java.* Concurrency; Practice & Experience, Vol. 9. No. 11. pp. 1225–1242. November 1997.

[15] V. Sarkar, *Partitioning and Scheduling Parallel Programs for Execution on Multiprocessors.* The MIT Press, 1989.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Parallel Performance Prediction for Numerical Codes in a Multi-Cluster Environment

Giuseppe Romanazzi, Peter K. Jimack
School of Computing
University of Leeds
LS2 9JT Leeds, United Kingdom
Email: {roman,pkj}@comp.leeds.ac.uk

***Abstract*—We propose a model for describing and predicting the performance of parallel numerical software on distributed memory architectures within a multi-cluster environment. The goal of the model is to allow reliable predictions to be made as to the execution time of a given code on a large number of processors of a given parallel system, and on a combination of systems, by only benchmarking the code on small numbers of processors. This has potential applications for the scheduling of jobs in a Grid computing environment where informed decisions about which resources to use in order to maximize the performance and/or minimize the cost of a job will be valuable. The methodology is built and tested for a particular class of numerical code, based upon the multilevel solution of discretized partial differential equations, and despite its simplicity it is demonstrated to be extremely accurate and robust with respect to both the processor and communications architectures considered. Furthermore, results are also presented which demonstrate that excellent predictions may also be obtained for numerical algorithms that are more general than the pure multigrid solver used to motivate the methodology. These are based upon the use of a practical parallel engineering code that is briefly described. The potential significance of this work is illustrated via two scenarios which consider a Grid user who wishes to use the available resources either (i) to obtain a particular result as quickly as possible, or (ii) to obtain results to different levels of accuracy.**

***Index Terms*—Parallel Distributed Algorithms; Grid Computing; Cluster Computing; Performance Evaluation and Prediction; Meta-Scheduling.**

## I. Introduction

As GRID computing becomes available as a practical commodity for computational science practitioners the need for reliable performance prediction becomes essential. In particular, when a variety of computational resources are available to a scientific research team they need to be able to make informed decisions about which resources to use, based upon issues such as the size of the problem they wish to solve, the turn-around time for obtaining their solution and the financial charge that this will incur. In order to make such decisions in a reliable way, it is necessary that they are able to predict the performance of their software across different combinations of these resources.

In this work we present a robust methodology for predicting the performance of parallel numerical multilevel software across different clusters (in terms of both processor and communications architectures) and across combinations of these clusters. The long term goal of this research is to model numerical software that requires a large computational cost, in a simple and cheap way using only few parallel runs across few processors.

Multilevel software (such as multigrid) has been selected for this work due to its growing importance in practical high performance computing software: as the maturity of multilevel algorithms continues to develop, it is able to provide excellent efficiency for very wide classes of problem [1], [2], [3], [4].

The methodology is first described and its predictive capability is then assessed for five different cluster configurations, using a typical parallel multigrid code. It is of course desirable that the predictive methodology proposed should be appropriate to the widest possible classes of numerical algorithms and the paper concludes with a discussion of these issues along with an illustrative example.

## II. Related Work

In previous work [5] we have begun to consider the use of simple (and cheap to implement) predictive models for the solution of certain classes of parallel multigrid codes when executed on distributed memory hardware. Whilst the results obtained in [5] are very encouraging, in this work we develop the ideas further in a number of significant ways.

1) A more general model for inter-processor communication is used which enables less-scalable communications patterns to be captured than previously. This is important when there are all-to-all communications at any point in the code and/or when the hardware does not scale well (e.g. Ethernet switching). The additional generality of this work also ensures that both blocking and non-blocking communication patterns can be reliably captured and modelled.
2) We extend our previous work to consider inter-, as well as intra-, cluster communications. Specifically, we now permit a single parallel job to be split across two entirely different clusters and the performance to be reliably predicted in advance.
3) In addition to reporting on the performance of our model as applied to benchmark multigrid codes, we also provide preliminary results which demonstrate that this performance is also achieved when applied to a practical multilevel engineering code [2].

978-83-60810-14-9/08/$25.00 © 2008 IEEE

There is a very substantial body of research into performance modelling [6] that varies from analytical models designed for a single application through to general frameworks that can be applied to many applications on a large range of high performance computing (HPC) systems. For example, in [7] detailed models of a particular application are built for a range of target HPC systems, whereas in [8] or [9] an application trace is combined with some benchmarks of the HPC system that is being used in order to produce performance predictions.

Both approaches have been demonstrated to be able to provide accurate and robust predictions, although each has its potential drawbacks: significant code specific knowledge being required for deriving the analytic models, whereas the trace approach may require significant computational effort. Moreover, in the former approach, when a different HPC system is used it would generally be necessary to change the model, adding new parameters for example. Instead, in the latter, we need to add or to find new benchmarks when a new code is used. Considering these limits, the choice between the two approaches can depend also on other factors. For example, when it is more important to predict the run-time of a large-scale application on a given set of systems, as opposed to comparing the performance of the systems in general, researchers (like those in the LANL group [7]) prefer to study deeply their application in order to obtain its own analytic model for the available set of HPC systems. On the other hand, when it is more interesting to compare performances of different machines on some real-applications, the latter approach is preferable; in that case different benchmark metrics can be used and convoluted with the application trace file.

Our approach lies between these two extremes. We use relatively simple analytic models (compared to the LogP model [10] for example), that are applicable to a general class of multigrid algorithms and then make use of a small number of simulations of the application on a limited number of CPUs of the target architecture in order to obtain values for the parameters of these models. Predictions as to performance of the application on larger numbers of processors may then be made.

As already indicated, our emphasis in this paper is to provide computational science practitioners with the tools to be able to make informed decisions concerning the Grid resources that they request. Indeed, the scenarios that we consider specifically relate to situations in which the Grid users are aware of which resources are immediately available (and can be reserved) or they are able to reserve resources at some future point in time. More generally however exactly the same information regarding the predicted execution time of a code on different resources, and different combinations of resources, is required by a Grid meta-scheduler for it to be able to work effectively. The job of such a scheduler is to evaluate different candidate resource sets and to select the “most suitable” resources for the execution of the application, e.g. [11]. It is with this in mind that our relatively light-weight approach to performance prediction becomes particularly attractive, since it is both simple and cheap to execute automatically.

There is of course a significant body of literature relating to performance models for large Grid environments. An excellent recent example is the research described in [12] which breaks the execution time of a parallel application into two parts, representing computation and communication costs, that are subsequently estimated for the target platform. Unlike our approach [12] is restricted to tasks that run on a single Grid resource, however the situation in which the load on the resource varies dynamically is included. Other researchers have also considered this situation, including the possible use of stochastic information to predict an application’s behaviour when there is contention for resources [13], [14]. In our work we assume that once a set of resources have been allocated they will be held exclusively by the application for the duration of the run or the reserved time slot, whichever is the shorter. Hence we do not consider this issue of contention here.

A variety of other papers on the subject of performance modelling in both dedicated and non-dedicated environments are described in [6] or [12], for example, so we do not repeat such reviews here. However, we finish this introduction by noting that the precise scheduling mechanism that is used for executing jobs on a Grid may have a significant influence on the performance of the prediction models themselves. Throughout this work we are focused on the situation where we are interested solely in the computational resources that are either available and ready to be used immediately, or the resources that may be reserved for use at some specified time in the future. All of the tests that were undertaken for this work were executed without the intervention of a scheduler. Instead, available resources were reserved and then the required jobs were launched.

## III. Parallel Numerical Software

Most numerical methods for the solution of partial differential equations (PDEs) are based upon the use of a spatial mesh for performing the discretization (as in finite difference, element, etc.), see for reference [15], [16].

Using parallel resources we are able to solve problems on finer grids than would be otherwise possible, so as to achieve greater accuracy. When the work per processor is kept constant, a parallel numerical software is considered efficient if there is only a slow increase in the execution time as the number of processors used grows. With multigrid algorithms, when the problem size is increased by a factor of $np$ then the solution time also grows by this factor, and so when solving on $np$ processors (instead of a single processor), the solution time should be unchanged. This would represent a perfect efficiency but is rarely achieved due to parallel overheads such as inter-processor communications and computations that are repeated on more than one processor.

In this research our aim is to be able to predict the execution time, including these overheads, of parallel numerical software running on $np$ processors. In some of the runs that follow we use more than one core per physical processor and for other runs we use a parallel architecture with a single core per physical processor. In each case we use the generic term processor to refer to each core or processor respectively. As suggested above, we restrict our attention to mesh-based PDE solvers, in this case considering a finite difference code with a series of non-blocking sends and receives in MPI, that solves a model PDE problem over a square two-dimensional domain (of size $N \times N$, say), see the multigrid code $m1$ in [5]. This domain is uniformly partitioned across the processors by assigning contiguous rows of the mesh to each processor in turn. In the case of a multigrid solver, the partitioning of the coarsest mesh ensures that all finer meshes are uniformly partitioned too (see [3], [17] for further details).

The top diagram in Fig. 1 illustrates a typical partition when $np = 4$. Each stage of the parallel numerical solver requires communications between neighbouring processors in order to update their neighbouring rows. This is typical in parallel numerical software of this type, e.g. [2], [3], [17].

## IV. The Predictive Model

The underlying observation upon which our model is based is that when we scale the size of our computational problem with respect to the number of processors used, the parallel overheads observed using just a small number of processors can describe the communication pattern for runs using a much larger number of processors. This occurs when the problem size per processor is kept fixed. In our methodology we therefore use parallel runs across few processors for predicting the performance of the parallel run across a large number of processors ($np$), with the same work assigned to each processor across all these runs. For convenience, here we define as “work per processor” the memory required by each processor: this is because the work load per processor in a multigrid code is proportional to the problem size assigned and therefore to the associated memory required by each processor.

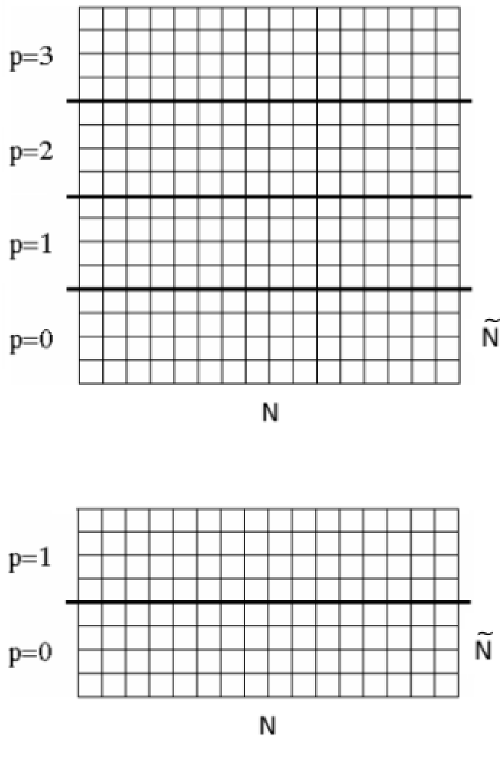


Fig. 1. Partitioning of a square mesh across four processors (top) and the equivalent problem considered on two processors (bottom).

The next basic assumption that we make is that the parallel solution time (on $np$ processors) may be represented as

$$T = T_{comp} + T_{comm}. \quad (1)$$

In (1), $T_{comp}$ represents the computational time for a problem of size $N \times \widetilde{N}$ on a single processor (where $\widetilde{N} = N/np$), and $T_{comm}$ represents all of the parallel overheads (primarily due to inter-processor communications).

The calculation of $T_{comp}$ is straightforward since this simply requires the execution of a problem of size $N \times \widetilde{N}$ on a single processor. Note that it is important that the precise dimensions of the problem solved on each processor in the parallel implementation are maintained for the sequential solve in order to obtain an accurate value for $T_{comp}$. This is because the memory access and contention patterns observed in the parallel runs (such as cache and multicore effects at the node-level) vary with respect to the geometrical dimensions of the memory allocated to each processor, and they can consequently influence the computational time measured.

The more challenging task is to model $T_{comm}$ in a manner that will allow predictions to be made for large values of $np$. Recall that our goal is to develop a *simple* model that will capture the main features of this class of numerical algorithm with just a small number of parameters that may be computed based upon runs using only a few processors. We present this model in (2) and then justify its simplicity in the remainder of the section.

$$T_{comm} = \alpha(np) + \gamma(np) \cdot work. \quad (2)$$

In (2) the term $work$ is used to represent the work on each processor, and is expressed in MBytes of the memory required, which is proportional to the computational cost. Also note that the length of the messages ($N$) does not appear in this formula since it is assumed that for a given size of target problem (e.g. a mesh of dimension $65536 \times 65536$) the size of the messages is known *a priori* (in this case, since the partition is by rows, the largest messages will be of length $65536$). Hence there is no need to include $N$ in the model as it is fixed in advance. This is the primary reason that the expression (2) can be so simple.

Furthermore, we will assume that the following relations also hold:

$$\begin{aligned} \alpha(np) &\approx c + d\log_2(np) \quad (3) \\ \gamma(np) &\approx \text{constant.} \quad (4) \end{aligned}$$

The justification for this model and the above assumptions are based upon our own empirical evidence gained using different parallel architectures. Two such illustrations are provided in Fig. 2 and Fig. 3. These show plots of overhead against work

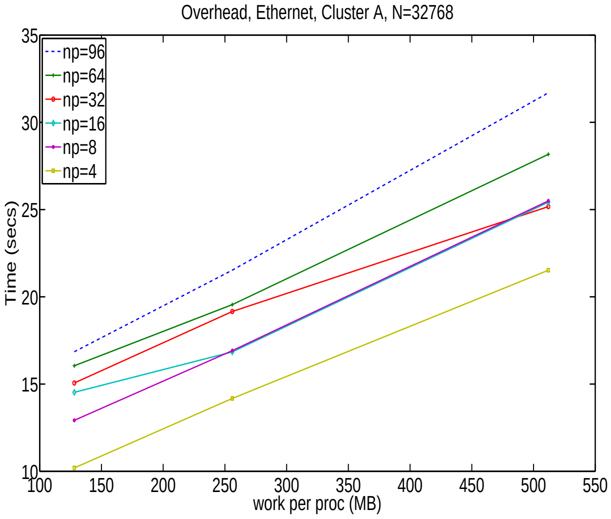


Fig. 2. Overhead ($T_{comm}$) associated to a fixed size of messages ($N$) using Fast Ethernet switching

for two different systems: one based upon a Fast Ethernet switching and the other based upon Myrinet. In each case we observe an almost linear growth in overhead with work, where the slope is approximately constant and there is an almost constant difference between graphs as $np$ is doubled. Note that the length of the messages is the same in all of these runs (see Fig. 1 for constant work with two different choices of $np$ and Fig. 4 for the same $np$ but half the work per processor).

In order to be able to use the model (2) it is necessary to evaluate the parameters $c$, $d$ and $\gamma$. These are determined using measurements taken for $np = 4$ and $np = 8$: $\gamma = \gamma(8)$ whilst $c$ and $d$ are obtained using a simple linear fit through the two data points.

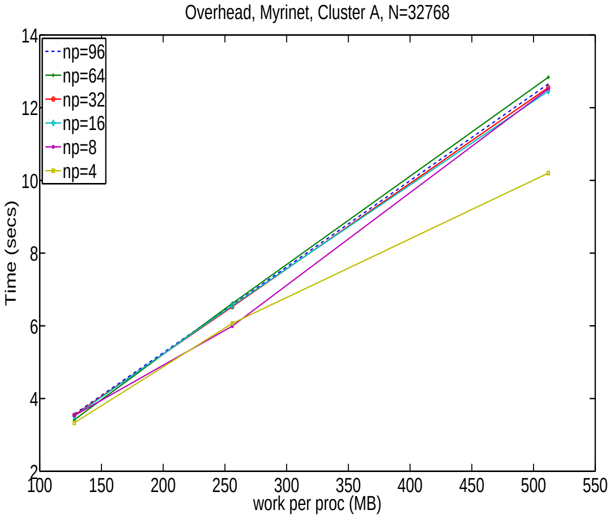


Fig. 3. Overhead ($T_{comm}$) associated to a fixed size of messages ($N$) using Myrinet switching.

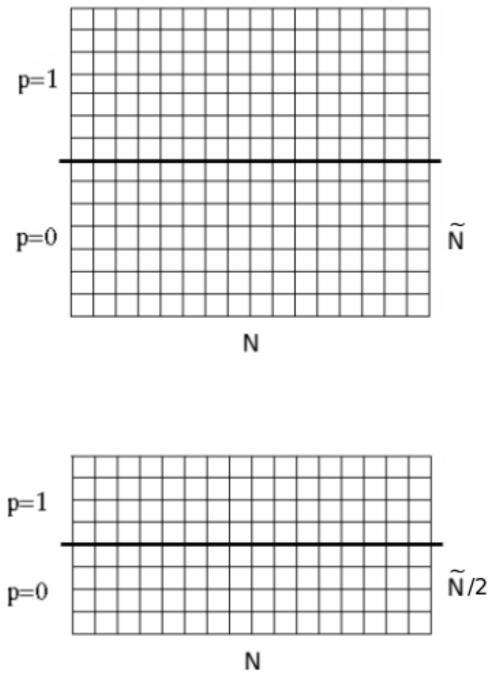


Fig. 4. Scaling the work per processor whilst maintaining the communication volume.

A summary of the overall predictive methodology is provided by the following steps. We define as $N \times N$ and $np$ the target problem size and number of processors respectively (i.e. we wish to predict a code's performance for these values). Also, let $\widetilde{N} = N/np$ and define $N \times \widetilde{N}$ to be the size of problem on each processor in the target configuration.

1) Run the code on a single processor with a fine grid of dimension $N \times \widetilde{N}$ and then with dimension $N \times \frac{\widetilde{N}}{4}$. In each case collect the computational time $T_{comp}$ and define as $work$ the memory allocated in the processor.
2) Run the code on $np0 = 4, 8$ processors, with a fine grid of dimension $N \times (np0 * \widetilde{N})$ and $N \times \left(np0 * \frac{\widetilde{N}}{4}\right)$. In each case collect the parallel time $T$ and then compute $T_{comm} = T - T_{comp}$.
3) Fit a straight line as in Eq. (2) (for both choices of $np = np0$) through the data collected in steps 1 and 2 to estimate $\alpha(np0)$ and $\gamma(np0)$.
4) Fit a straight line as in Eq. (3) through the points $(2, \alpha(4))$ and $(3, \alpha(8))$ to estimate $c$ and $d$: based upon Eq. (3) now compute $\alpha(np)$ for the required choice of $np$.
5) Use the model in Eq. (2) to estimate the value of $T_{comm}$ for the required choice of $np$ (using the values $\gamma(np) = \gamma(8)$ and $\alpha(np)$ determined in steps 3 and 4 respectively).
6) Combine $T_{comm}$ from step 5 with $T_{comp}$ (determined in step 1, with finest size $N \times \widetilde{N}$) to estimate $T$ as in Eq. (1).

In the parallel runs described in step 2, we use messages at all levels with lengths equal to those used in the parallel run that we are interested to predict. As we show in the next section, this permits us to describe accurately the communication patterns at all mesh levels of the multigrid code.

## V. Numerical Results

The approach described in the previous section is now used to predict the performance of a typical numerical code running on two different clusters, either individually or together.

### A. The White Rose Grid

The White Rose Grid is a collaborative project involving the Universities of Leeds, Sheffield and York [18]. In these tests we make use of two clusters on this Grid.

- Cluster A (White Rose Grid Node 2) is a cluster of 128 dual processor nodes, each based around 2.2 or 2.4GHz Intel Xeon processors with 2GBytes of memory and 512 KB of L2 cache. Either Myrinet or Fast Ethernet switching may be used to connect the nodes.
- Cluster B (White Rose Grid Node 3) is a cluster of 87 Sun microsystem dual processor AMD nodes, each formed by two dual core 2.0GHz processors. Each of the $87 \times 4 = 348$ batched processors has L2 cache memory of size 512KB and access to 8GBytes of physical memory. Again, both Myrinet and Fast Ethernet switching are available.

In addition to running jobs on either cluster, using either switching technology, it is also possible to run a single parallel application across both clusters together (using Fast Ethernet only).

Because users of clusters A and B do not get exclusive access to their resources some variations in the execution time of the same parallel job can be observed across different runs. A simple way to reduce such effects in the predictive methodology is to take average timings on a limited number of runs. However, this approach alone is not sufficient since specific hardware features must also be accounted for.

For cluster A, for example, there are 75 2.4GHz and 53 2.2GHz dual processors, hence it is necessary to ensure that all runs used in the parameter estimation phase make use of at least one slower processor. This is because if only the faster processors are used to estimate $T_{comm}$ and $T_{comp}$, then the resulting model will under-predict solution times on large numbers of processors (where some of the processors will be 2.2GHz rather than 2.4GHz). Similarly, on the multicore cluster B, care needs to be taken to account for this architectural feature. For example, all of the sequential runs are undertaken using four copies of the same code: each running on the same (four-core) node. Again, this decision is made bearing in mind the situation that will exist for a large parallel run in which all the available cores in a node are likely to be used. Moreover on this cluster the $8$ core runs, distributed as two full nodes, are able to catch both intra- and inter-node communications, see [5] for further details. This strategy permits to reproduce the effects [19] of the memory contention at the node-level in a multi-core architecture.

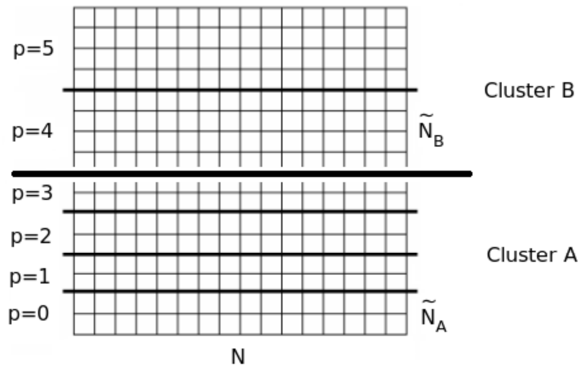


Fig. 5. Example partitions by rows of a fine square mesh across two clusters: A and B.

### B. Methodology for Inter-Cluster applications

As mentioned above, it is also possible to run a single job across both clusters using Fast Ethernet switching. Fig. 5 illustrates a typical partition, for which the work per processor may be different on each cluster. In this example a target configuration with $np_A$ processors on cluster A (each working with a sub-mesh of size $N \times \widetilde{N}_A$) and $np_B$ processors on cluster B (each working with a sub-mesh of size $N \times \widetilde{N}_B$) is assumed. In order to predict the overall solution time for such a multi-cluster run we make the assumption that the inter-cluster communication costs, whilst greater than those within each cluster, will generally be negligible compared to the inevitable imbalance of execution times between the clusters. Hence our methodology is to use the approach of the previous section to predict $T_A$ for the problem of size $N \times (np_A * \widetilde{N}_A)$ assigned to the $np_A$ processors of cluster A and $T_B$ for the problem of size $N \times (np_B * \widetilde{N}_B)$ on the $np_B$ processors of cluster B. We then take the simple estimate

$$T = \max (T_A, T_B). \tag{5}$$

### C. Results

We have tested our models for a range of problems with five different cluster architectures and present a selection of typical results in Tables I and II below. These tables are focused around two potential applications of the predictions within a Grid environment, which we refer to here as scenarios. However the key observation that wish to we make here is the consistent accuracy of the predictions when compared to the actual run times that have subsequently been computed.

### Scenario 1

In this scenario, it is assumed that a problem of a particular size must be solved and that two clusters are scheduled to be partially available, with $np_A$ and $np_B$ processors free on clusters A and B, respectively. Specifically, we consider the

TABLE I
MEASUREMENTS AND PREDICTIONS (BOTH QUOTED IN SECONDS) FOR SCENARIO 1.

| procs | $np_A = 64$ | $np_A = 64$ | $np_B = 32$ | $np_B = 32$ | $(np_A, np_B) = (64, 32)$ |
|---|---|---|---|---|---|
| switching | Ethernet | Myrinet | Ethernet | Myrinet | Ethernet |
| size | $65536^2$ | $65536^2$ | $65536^2$ | $65536^2$ | $65536^2$ |
| mem. per core/proc | 2GB | 2GB | 4GB | 4GB | ( 1GB, 2GB ) |
| measurement | 1703.9 | 1014.9 | - | - | 1104.6 |
| prediction | 1715.7 | 983.9 | - | - | 1044.4 |
| $\lvert$error$\rvert$ | 0.69% | 3.05% | - | - | 5.45% |

case $np_A = 64$, $np_B = 32$ for a target problem size of $N \times N$ with $N = 65536$, see Table I. The memory requirement across some different combinations of processors is shown in the fourth row. The columns entitled "$np_A = 64$" show two sets of predicted and actual results using 64 processors on cluster A: based upon Ethernet and Myrinet switching respectively. The columns entitled "$np_B = 32$" are empty, reflecting the fact that insufficient memory is available to execute a job of this size on 32 cores of cluster B alone. The final column shows predicted and actual results when the job is split equally between the two clusters (using 64 and 32 cores on clusters A and B respectively). In all cases, the model is demonstrated to provide excellent predictions to the actual measured run times.

The purpose of this scenario is to illustrate a situation in which the user wishes to decide which of a number of combinations of available resources will deliver the required answer in the shortest time. Here the user is able to determine whether it will be better to use 64 processors of cluster A alone or a combination of these processors along with the 32 available cores of cluster B. In this particular case, if only Ethernet is available then the latter approach is faster whereas the former would be better if Myrinet is available on cluster A. Assuming that pricing information is available to the user (based upon a different rate per cpu hour on each cluster) it is also possible to predict the financial cost of each option in advance.

Other combinations of processors and job partition may be assessed in the same manner according to what resources are scheduled to be available at any given time. For example if there are an equal number of processors available on cluster A and B then it is likely to be desirable to give the faster cluster more than half of the computational domain to work with.

### *Scenario 2*

In the second scenario that we present, a user wishes to consider solving a problem with different levels of mesh resolution. That is, given two Grid resources that are simultaneously available, they can either choose to solve on the larger of these two resources or else they can make use of both resources together in order to solve a problem with even more unknowns (using the memory of both resources together). In the latter case it will clearly be possible to get more resolution but the user may wish to know how much extra this will cost, and will therefore need a reliable estimate of the solution time for each alternative.

Table II shows five different predictions, along with the corresponding measured runs times, for different cluster configurations. It is assumed that up to 32 processes are available on either of cluster A or B, or on each of them together. In the single-cluster cases the largest problem that may be solved for which $N$ is a power of 2 is $32768 \times 32768$, which corresponds to approximately 1GByte of memory per processor. By combining the two clusters however it is possible to obtain a solution with $N = 65536$. As for the previous table of results, it is again clear that the predictions obtained using the methodology described in this paper prove to be remarkably robust given their simplicity.

The significance of this scenario is that, in a Grid computing environment, our predictions provide users with the tools required to make an informed decision as to what resources they wish to request. By combining resources from two different clusters it is possible to solve a problem with greater mesh resolution however the cost of doing so may be substantial. In this specific case if, for example, cluster A is charged at 1 unit per cpu hour and cluster B is charged at 2 units per cpu hour, then the financial cost (both predicted and actual) of obtaining the greater resolution by using both clusters would be approximately 10 times the cost of the $32768 \times 32768$ resolution run using cluster B with Myrinet.

Note that, as with scenario 1, it is possible to consider other combinations of available processors using this same approach. Unlike scenario 1 however, in this case our focus is on maximizing the amount of memory available rather than minimizing the run time required.

## VI. DISCUSSION

In this paper we have proposed a simple methodology for predicting the performance of parallel numerical codes within a multi-cluster environment. The philosophy upon which this methodology is based is to produce a general empirical model that involves a minimum number of parameters, and then to determine appropriate values for these parameters for any given combination of code and hardware resources. These parameter values are determined based upon the characteristics of the code when it is executed on much smaller numbers of processors than are ultimately required. This allows resources that are not currently available to be reserved for future execution based upon the predicted need. Results presented in the previous section demonstrate that the methodology is both robust and accurate across five different combinations of parallel architecture for a given multigrid code. Furthermore,

TABLE II
MEASUREMENTS AND PREDICTIONS (BOTH QUOTED IN SECONDS) FOR SCENARIO 2.

| procs | $np_A = 32$ | $np_A = 32$ | $np_B = 32$ | $np_B = 32$ | $(np_A, np_B) = (32, 32)$ |
|---|---|---|---|---|---|
| switching | Ethernet | Myrinet | Ethernet | Myrinet | Ethernet |
| size | $32768^2$ | $32768^2$ | $32768^2$ | $32768^2$ | $65536^2$ |
| mem per core/proc | 1GB | 1GB | 1GB | 1GB | ( 2GB, 2GB ) |
| measurement | 776.7 | 628.3 | 444.4 | 281.0 | 1645.5 |
| prediction | 739.2 | 628.6 | 451.9 | 259.5 | 1686.0 |
| $\|\text{error}\|$ | 4.83% | 0.05% | 1.69% | 7.65% | 2.46% |

two different Grid scenarios have been considered, for which the performance prediction is of clear practical value.

Although the results presented in this work have been computed without the aid of any automatic scheduling software, it is clear that the performance prediction capability that has been demonstrated is of great potential value to Grid middleware and meta-schedule developers. When applications are submitted to a Grid, the scheduler needs accurate information regarding the potential performance of those applications on different resource combinations in order to be able to make optimal choices regarding the allocation of jobs to resources. We hope to explore this feature of our work further in future research. In order to be of maximum value however it will be necessary to demonstrate the generality of our approach to other numerical software.

In addition to the standard linear multigrid code that has been used for testing here, the methodology can be shown to extend to other parallel multilevel software too. Examples from our current work include the simulation of the spreading of fluid droplets [3] and the simulation of nonlinear lubrication problems involving fluid-structure interaction [2]. Details of the practical application to these engineering problems on a single cluster form the subject of another publication [20], however sample results are included here as evidence of the generality of our approach. Table III illustrates timings and predictions for the code described in [2], where we use the same methodology as described in this paper, based upon the separate prediction of $T_{comp}$ and $T$. In this case the code has additional components to the pure multigrid codes used for the rest of this paper and the work no longer scales linearly with memory. Nevertheless, as Table III clearly shows, provided this is taken into account the basic methodology that we propose again provides excellent predictions.

In addition to applying and testing our methodology to practical scientific codes in 2-d, one of the next steps that we wish to undertake is the application in 3-d. When the same linear partition of the problem is used then it is expected that the approach will be equally successful however further developments are required in order to deal with more general partitioning strategies. It is also our intention to assess the quality of the methodology when applied to other numerical schemes than the multilevel finite difference and finite element codes so far investigated. Candidates for a successful application includes other structured approaches such as Lattice-Boltzmann simulations [21].

## ACKNOWLEDGEMENTS

We are very grateful to Drs. Chris Goodyer and Jason Wood for their valuable input to this research which is supported by EPSRC grant EP/C010027/1. We also thank anonymous referee for their comments concerning Grid meta-scheduling.

## REFERENCES

[1] R. E. Bank, and M. J. Holst, "A New Paradigm for Parallel Adaptive Meshing Algorithms," *SIAM Review* vol. 45, 2003, pp. 292–323.

[2] C. E. Goodyer, and M. Berzins, "Parallelization and scalability issues of a multilevel elastohydrodynamic lubrication solver," *Concurrency and Computation,* vol. 19, 2007, pp. 369–396.

[3] P. H. Gaskell, P. K. Jimack, Y. Y. Koh, and H. M. Thompson, "Development and application of a parallel multigrid solver for the simulation of spreading droplets," *Int. J. Num. Meth. Fluids*, vol. 56, 2008, pp. 979–1002.

[4] P. Ladeveze, A. Nouy, and O. Loiseau, "A multiscale computational approach for contact problems", *Comput. Meth. Appl. Mech. Engrg.*, vol. 191, 2002, pp. 4869-4891.

[5] G. Romanazzi, and P. K. Jimack, "Parallel performance prediction for multigrid codes on distributed memory architectures", in *High Performance Computing and Communications (HPCC-07)*, ed. R. Perrott et al. (LNCS 4782, Springer), 2007, pp. 647–658.

[6] S. Pllana, I. Brandic and S. Benkner, "A survey of the state of the art in performance modeling and prediction of parallel and distributed computing systems", *Int. J. Comput. Intel. Res. (IJCIR),* vol. 4, 2008, pp. 17–26.

[7] D. J. Kerbyson, H. J. Alme, A. Hoisie, F. Petrini, H. J. Wasserman and M. Gittings, "Predictive performance and scalability modeling of a large-scale application", in *Proceedings of SuperComputing 2001*, 2001.

[8] G. Rodriguez, R. M. Badia, and J. Labarta, "Generation of simple analytical models for message passing", in *Euro-Par 2004 Parallel Processing*, ed. M. Danelutto et al. (LNCS 3149, Springer), 2004, pp. 183–188.

[9] L. Carrington, M. Laurenzano, A. Snavely, R. Campbell and L. Davis, "How well can simple metrics represent the performance of HPC applications?", in *Proceedings of SuperComputing 2005*, 2005.

[10] D. E. Culler, R. M. Karp, D. A. Patterson, A. Sahay, K. E. Schauser, E. Santos, R. Subramonian and T. von Eicken, "LogP: towards a Realistic Model of Parallel Computation", *SIGPLAN Not.*, vol. 28, 1993, pp. 1–12.

[11] A. Petitet, S. Blackford, J. Dongarra, B. Ellis, G. Fagg, K. Roche and S. Vadhiyar, "Numerical libraries and the grid: The grads experiments with scalapack", *J. of High Performance Applic. and Supercomputing*, vol. 15, 2001, pp. 359–374.

[12] H. A. Sanjay and S. Vadhiyer, "Performance modeling of parallel applications for grid scheduling", *J. Parallel Dist. Comput.*, vol. 68, 2008, pp. 1135–1145.

[13] J. Schopf and F. Berman, "Performance prediction in production environments", in *Proceedings of 12th International Parallel Processing Symposium*, Orlando, USA, 1998.

[14] J. Schopf and F. Berman, "Using stochastic information to predict application behaviour on contended resources", *Int. J. Found. Comput. Sci.*, 12, 2001, pp. 341364.

[15] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*, Elsevier, 2002.

[16] S. S. Rao, *The Finite Element Method in Engineering,* Butterworth-Heinemann, 2005.

TABLE III
MEASUREMENTS AND PREDICTIONS (BOTH QUOTED IN SECONDS) FOR THE COMPUTATIONAL ENGINEERING APPLICATION SOFTWARE DESCRIBED IN [2], BASED UPON OUR PREDICTIVE METHODOLOGY ASSESSED FOR CLUSTERS A AND B USING MYRINET SWITCHING.

| procs cluster A<br>size | $np_A = 64$<br>$16385 \times 8193$ | $np_A = 128$<br>$16385 \times 16385$ |
|---|---|---|
| measurement | 1074.86 | 1260.24 |
| prediction | 1051.31 | 1242.86 |
| $\|\text{error}\|$ | 2.91% | 1.38% |

| procs cluster B<br>size | $np_B = 64$<br>$16385 \times 8193$ | $np_B = 128$<br>$16385 \times 16385$ |
|---|---|---|
| measurement | 908.44 | 1124.19 |
| prediction | 904.39 | 1107.79 |
| $\|\text{error}\|$ | 0.44% | 1.45% |

[17] S. Lang, and G. Wittum, “Large-scale density-driven flow simulations using parallel unstructured grid adaptation and local multigrid methods”, *Concurrency and Computation: Practice and Experience,* vol. 17, 2005, pp. 1415–1440.

[18] P. M. Dew, J. G. Schmidt, M. Thompson, and P. Morris, “The White Rose Grid: practice and experience,” *in Proceedings of the 2nd UK All Hands e-Science Meeting,* ed. S.J. Cox, EPSRC, 2003.

[19] M. D. McCool, “Scalable programming models for massively multicore processors”, *Proceedings of the IEEE,* vol. 96, 2008, pp. 816-831.

[20] G. Romanazzi, P. K. Jimack, and C. E. Goodyer, “Reliable performance prediction for parallel scientific software in a multi-cluster grid environment”, in *Proceedings of the Sixth International Conference on Engineering Computational Technology,* Civil-Comp Press, 2008, to appear.

[21] A. R. Davies, J. L. Summers, and M. C. T. Wilson, “Simulation of a 3-D lid-driven cavity flow by a parallelised lattice Boltzmann method”, in *Parallel Computational Fluid Dynamics: new Frontiers and Multi-Disciplinary Applications*, 2003, pp. 265-271.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 475–480

ISBN 978-83-60810-14-9
ISSN 1896-7094

# On the Robustness of the Soft State for Task Scheduling in Large-scale Distributed Computing Environment

Harumasa Tada
Faculty of Education
Kyoto University of Education
1, Fujinomori-cho, Fukakusa, Fushimi-ku,
Kyoto 612-8522, Japan
Email: htada@kyokyo-u.ac.jp

Makoto Imase, Masayuki Murata
Graduate School of Information Science
and Technology
Osaka University
1-5, Yamadaoka, Suita, Osaka 565-0871, Japan
Email: {imase,murata}@ist.osaka-u.ac.jp

***Abstract*—In this paper, we consider task scheduling in distributed computing. In distributed computing, it is possible that tasks fail, and it is difficult to get accurate information about hosts and tasks. WQR (workqueue with replication), which was proposed by Cirne et al., is a good algorithm because it achieves a short job-completion time without requiring any information about hosts and tasks. However, in order to use WQR for distributed computing, we need to resolve some issues on task failure detection and task cancellation. For this purpose, we examine two approaches—the conventional task timeout method and the soft state method. Simulation results showed that the soft state method is more robust than the task timeout method.**

## I. Introduction

DISTRIBUTED computing, in which large-scale computing is performed using the idle CPU times of many PCs, has attracted considerable attention recently. The jobs executed in distributed computing comprise many tasks. These tasks are allocated to PCs and are processed in parallel. Some well-known active distributed computing projects are SETI@home[1] and distributed.net[2].

So far, distributed computing has been mainly used in the limited area of scientific computing for analysis of protein folding, climate simulations, nuclear physics, etc. In these type of applications, the jobs are usually so large that they often take several months to be completed. For such large jobs, the effect of task scheduling on the job completion time is relatively small. Therefore, an efficient task scheduling algorithm is not that important in traditional distributed computing projects.

In the future, it is expected that distributed computing will be applied to various types of large-scale computing applications such as analysis of DNA, data mining, simulations of atmospheric circulation or ocean circulation, structural and stress analysis and fluid analysis of the air resistance of cars or planes and the water resistance of ships. These type of applications require many medium-sized jobs that take several hours or days to complete. The job completion time of such medium-sized jobs are affected by task scheduling.

In this paper, we discuss task scheduling in distributed computing.

Task scheduling in distributed computing has two main goals. The first goal is to minimize the job completion time. In order to complete a job, all tasks that are executed on various hosts should be completed. The delay in a single task can affect the completion time of the whole job. Therefore, the scheduler has to monitor the task execution at each host and perform appropriate actions in response to the change in a situation. The second goal is to minimize the wastage of CPU cycles. In distributed computing, it is common to replicate tasks to achieve good performance. However, using task replicas results in a wastage of CPU cycles. In traditional distributed computing projects, this wastage of CPU cycles has been tolerated because such CPU cycles otherwise go into idle cycles. However, the wastage of CPU cycles implies the wastage of electric power if PCs have power-saving function. Considering recent trend of energy saving, the importance of the reduction in wastage of CPU cycles in distributed computing is increasing.

Task scheduling is not a new problem. It has been studied in the area of parallel computers or clusters. However, the distributed computing systems targeted in this paper are different from parallel computers or clusters. The characteristics of distributed computing are listed as follows:

- Hosts are heterogeneous and autonomous[3].
- It is difficult to obtain good information about the hosts[4].
- Hosts are often behind NATs (network-address translations) or firewalls[5].
- Hosts are frequently turned off by users[5].

In BOINC[5], the well-known distributed computing platform, task scheduling is performed on the basis of the information regarding the processing power of each host and the estimation of the processing time of each task. When this information does not reflect the actual performance, the job completion time deteriorates.

In this paper, we investigate a task scheduling method known as WQR (workqueue with replication)[6], which was

978-83-60810-14-9/08/$25.00 © 2008 IEEE

originally proposed for heterogeneous Grid environments. The main feature of WQR is that it does not require any kind of information about the hosts or tasks. WQR has the same performance as the existing scheduling method used, which requires informations on the hosts and tasks. For this reason, the use of the WQR method in distributed computing appears promising. The drawback of WQR is the wastage of CPU cycles. In WQR, some CPU cycles are wasted because tasks are replicated and processed by multiple hosts.

There are some issues to consider when WQR is applied to distributed computing. First, the original WQR method does not take into account task failures and therefore it is possible that the job will not be completed if task failures occur frequently. In distributed computing, however, task failures are not exceptional because the hosts are frequently turned off by users. Second, the original WQR method assumes that the scheduler is able to cancel executing tasks by sending messages to the hosts. However, in distributed computing, it is difficult for the scheduler to send messages to the hosts because of NATs or firewalls. Therefore, in order to use the WQR method in distributed computing, it is necessary to add additional mechanisms to enable task failure detection and task cancellation.

A common approach to solve these issues is to set a timeout for task execution. Most existing distributed computing platforms use this approach[7]. If a scheduler does not receive the result of an allocated task before the timeout, the scheduler creates another replica of the task and allocates it to another host. We call this method as the task timeout method. The task timeout method ensures the completion of all tasks of the job. The problem in this approach is the determination of the appropriate timeout value. The appropriate timeout value depends on the average task execution time and therefore cannot be determined uniquely. The wrong timeout value may cause a delay in job completion and wastage of CPU cycles.

Another approach involves the use of the soft state protocol, which improves the robustness of distributed systems[8]. In this approach, hosts and schedulers exchange messages periodically in order to monitor each other's states. Messages are sent in a best-effort (unreliable) manner[9], that is, they are not retransmitted if they are lost.

In this study, we used the task timeout method and the soft state method with WQR and compared their performances through simulations. The simulation results showed that the soft state method had a better performance than the task timeout method.

The rest of this paper is organized as follows. In section II, we define the distributed computing model and the terms used in this study. In section III, we provide an overview of WQR. In section IV, we explain the issues in applying WQR to distributed computing. In section V, we describe two approaches to deal with these issues. In section VI, we evaluate the performance of these approaches through simulations. Finally, in section VII, we conclude the paper.

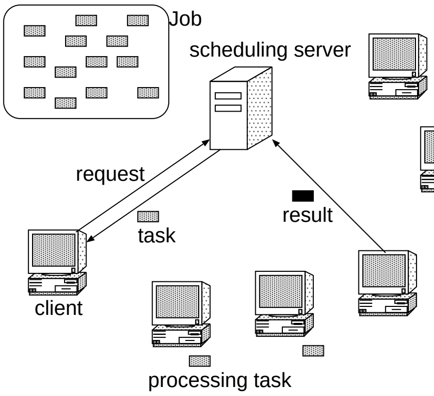


Fig. 1. Model of the distributed computing environment

## II. The distributed computing environment model

Distributed computing involves performing large-scale computations using the idle CPU cycles of many PCs in a network. The participating PCs are known as *hosts*. We use the term *users* to refer to the people using the hosts. A *task* is a unit of scheduling that consists of a program and input data. A *job* is defined as a set of tasks.

Task scheduling involves the allocation of tasks to hosts along with the allocation of the order of execution. Like most distributed computing projects, we consider Bag-of-Tasks applications in which tasks are completely independent[4].

Fig. 1 shows the model of the distributed computing environment considered in this paper. It consists of one scheduler and many hosts.

When the scheduler receives a job, it allocates the tasks in the job to hosts. The hosts receive the tasks from the scheduler and send back the results after task completion. We assume that the transfer of tasks and results are performed in a reliable manner using retransmissions.

However, hosts sometimes crash because of hardware failures or shutdowns by users. When hosts crash, their tasks also fail. Task failures are not notified explicitly to the scheduler.

In this paper, we distinguish between task failure and task cancellation. The former implies that task execution is stopped unexpectedly due to crashing of the host. The latter implies that the task execution is stopped as the task is considered unnecessary by the scheduler.

## III. Workqueue with Replication

Workqueue with replication (WQR)[6] is a task scheduling algorithm originally proposed for large-scale distributed systems such as Grids. The main feature of WQR is that it does not require any kind of information about the hosts or tasks. The performance of WQR is equivalent to that of FPLTF[10] or Sufferage[11], which require information about the hosts and tasks[6].

The WQR algorithm uses task replication to achieve a good performance. The scheduler chooses tasks in an arbitrary order and sends them to the hosts. When a host completes its task, it sends the result back to the scheduler and receives a new task from the scheduler. This scheme continues until all the tasks in the job are allocated. In the simple Workqueue algorithm, hosts that complete their tasks become idle. In WQR, however, these hosts are allocated replicas of tasks that are still running. When a task replica finishes at any host, its other replicas are cancelled.

Using task replication, WQR can improve the performance of a distributed computing by reducing the delay in completion of tasks allocated to slow/busy hosts. Task replication increases the possibility that at least one replica is allocated to a fast/idle host.

However, to avoid wastage of CPU cycles, there is a predefined limit on the number of task replicas. Tasks are replicated until the number of replicas reaches this predefined limit.

## IV. Issues in application of WQR to distributed computing

The concept of WQR is very desirable in a distributed computing environment in which it is difficult to obtain accurate information about the performance of hosts. However, the original WQR algorithm is not recommended for use in a distributed computing environment. When the WQR method is applied to distributed computing, there are some issues to consider.

### A. Detection of task failure

When a task fails, the scheduler should detect the failure and reschedule the failed task to a different host. However, it is very difficult for the scheduler to detect task failures without any notification.

In the original WQR method, the number of task replicas that can be created is limited. If all the replicas of a task fail and these failures are not detected by the scheduler, the task can never get completed. Such a situation can be avoided if the number of task replicas is unlimited. However, this increases the wastage of CPU cycles[6].

### B. Explicit task cancellation

In the original WQR method, when a task replica is completed by any host, the executions of its other replicas are cancelled in order to reduce the wastage of CPU cycles.

However, in distributed computing, it is difficult for the scheduler to cancel the tasks executing on hosts. This is because in a distributed computing environment, the hosts are generally behind NATs or firewalls. In this situation, communications are always initiated by the hosts and not by the scheduler. The scheduler cannot send any messages unless the hosts open a connection to the scheduler. Therefore, even though the scheduler receives the results of the completed task, it cannot send messages to the other hosts to cancel the replicas.

## V. Extensions of WQR

We consider two approaches to improve the WQR method, i.e., the task timeout method and the soft state method, in order to deal with the issues discussed in section IV.

### A. Task timeout method

In BOINC, the task timeout method, which has an upper limit for task execution time, is used.

The scheduler allocates a task to a host and sets the timeout value on the timer. If the scheduler does not receive the result of the task from the host before the timeout period elapses, it reschedules the task to another host. This method ensures the completion of all tasks of the job.

A problem with the timeout method is the determination of an appropriate timeout value. The appropriate timeout value depends on the average task execution time. An inappropriate timeout value delays job completion and wastes many CPU cycles. For example, if the timeout value is too small, the scheduler creates redundant replicas of tasks that are still running, while if the timeout value is too large, there is a delay in the detection of failed tasks by the scheduler.

The explicit task cancellation mentioned in section IV-B is impossible in the task timeout method. All task replicas are executed until they have completed or failed.

### B. Soft state method

The soft state method was originally proposed for state management of communication protocols[12]. It is characterized by periodic refreshing of information and the initialization of information by timeout. Lui et al. showed that the use of the soft state method makes communication protocols highly robust[8].

In the soft state method, the scheduler and hosts exchange information with each other. Hosts that are executing tasks send refresh messages to the scheduler periodically. On receiving the refresh message, the scheduler sends a reply message to the host. Messages are sent in a best-effort (unreliable) manner, that is, they are not retransmitted if they are lost. If the scheduler does not receive a refresh message before the message timeout, it reschedules the corresponding task to another host. On the other hand, if the host does not receive a reply message before the message timeout, it aborts its task and sends a request for a new task to the scheduler.

The main feature of this method is that the scheduler and the hosts are aware of each other's states due to the periodic exchange of messages. On receiving refresh messages from the hosts, the scheduler confirms that the allocated tasks are running normally. The reply messages from the scheduler to the hosts confirm that the scheduler is still waiting for the results of the tasks, that is, the tasks are not cancelled. Thus, the soft state method can perform explicit task cancellation, which cannot be achieved with the task timeout method. By stopping the flow of refresh messages, the scheduler can inform the host that the task is no longer necessary.

However, problems can arise if message losses occur frequently. If refresh messages are lost consecutively, the scheduler regards the corresponding task as failed and creates redundant replicas of the task. If reply messages are lost consecutively, the host regards its task replica as cancelled and aborts the necessary task replica. This false task abortion is another cause of task failure.

In order to exchange messages with the scheduler, hosts should always be connected to the network. However, since the sizes of refresh and reply messages are very small, the network load added by these messages is negligible compared to the tasks and their results.

## VI. Performance Evaluation

We evaluated the performance of the two methods mentioned in section V through simulations.

We assumed that the network transfer times were negligible because our target applications were CPU bound. As the performance criteria, we considered the job completion time and the number of wasted CPU cycles The job completion time is the time between the start of the first task and the completion of the last task. The wasted CPU cycles are the sum of all the CPU cycles that are used for executing task replicas that did not contribute to the final result. Such task replicas include cancelled replicas, failed ones, and redundantly completed ones.

In the rest of this section, we will refer to the task timeout method and the soft state method as the TT and SS methods, respectively.

### A. Simulation settings

In our simulations, the scheduler runs a single job. Therefore, all hosts execute tasks of the same job. Further, each host executes only one task at a time.

The processing power of each host is taken from a uniform distribution $U(1, 7)$ and therefore the average processing power of the hosts is 4. The total processing power of all the hosts in the system is fixed to 1000.

The size of a task is defined as the processing power required to process the task. For example, a task with a size of 10 is processed in 5 s by a host with a processing power of 2. The average task size (denoted as $\mu$) is 10000 (large tasks) or 100 (small tasks). The size of each task is an integer taken from a normal distribution with a standard deviation of $0.4\mu$. In both cases, the total size of all tasks included in a job is fixed as 1000000.

The number of task replicas is limited to 4 on the basis of the results of the study by Cirne et al.[6], which states that the performance of WQR with a limit of 4 is close to that when the number of task replicas is unlimited.

The task failure rate is defined as the inverse of the MTBF (mean time between failures) of tasks. All tasks have the same failure rate regardless of their size. This reflects the fact that large tasks stay in the hosts for a long time and therefore they are more likely to be involved in host crashes.

The message loss rate is defined as the ratio of lost messages to total messages sent. If the message loss rate is 0.1, it implies that 10% of the messages are lost.

For the TT method, the simulations were performed by changing the task timeout value. We used three timeout values $e_s$, $2e_s$, and $4e_s$. $e_s$ is the estimated average task execution time given by $e_s = \mu/p$, while $p$ is the average processing power ($p = 4$, as mentioned in section VI-A). For example, if the average task size is 10000, then the task timeout values are 2500, 5000, and 10000. In the following graphs, “TT-$X$” indicates a task timeout method with timeout value $X$.

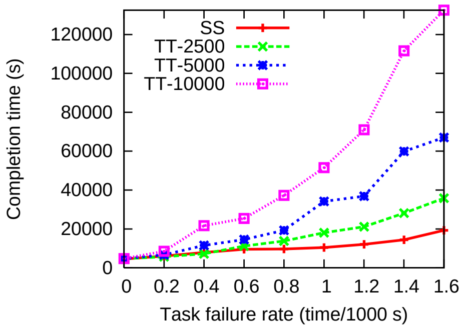


Fig. 2. Plot of the job completion time ($\mu = 10000$)

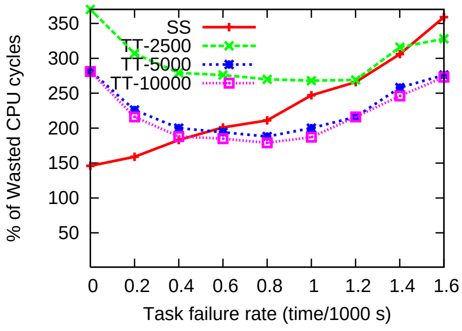


Fig. 3. Plot of the wasted CPU cycles ($\mu = 10000$)

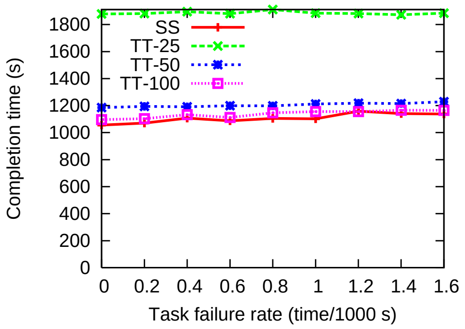


Fig. 4. Plot of the job completion time ($\mu = 100$)

In the simulations of the SS method, refresh messages were sent every 10 s and the message timeout value was set to 100(s). This implies that 10 or more consecutive message losses resulted in rescheduling or abortion of tasks, as mentioned in section V-B.

### B. Simulation results

For each parameter setting, we performed 10 simulations using different jobs. The presented results are the average of the results of the 10 simulations.

*1) Effect of task failures:* The task failure rate was changed for all simulations. The message loss rate was fixed at 0.

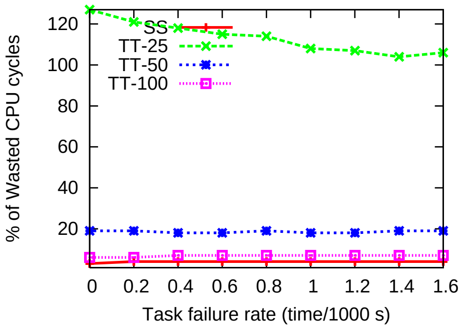


Fig. 5. Plot of the wasted CPU cycles ($\mu = 100$)

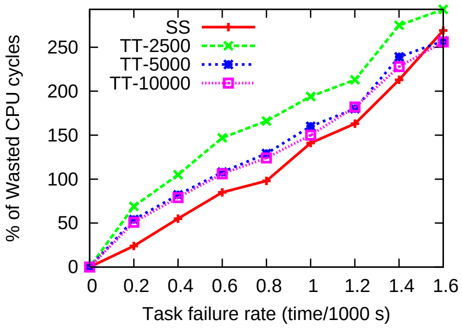


Fig. 6. Plot of the CPU cycles wasted by failed tasks ($\mu = 10000$)

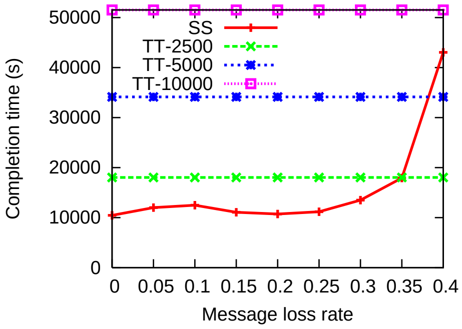


Fig. 7. Plot of the job completion time ($\mu = 10000$)

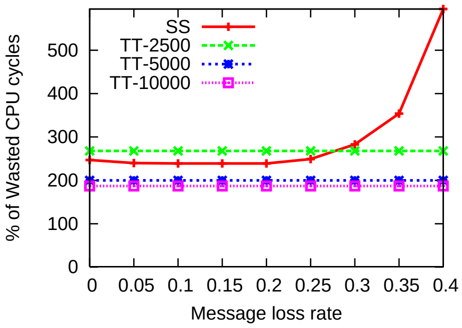


Fig. 8. Plot of the wasted CPU cycles ($\mu = 10000$)

Fig. 2 shows the job completion time when the average task size $\mu$ is large. It can be observed that the job completion time of the SS method is short and the impact of the task failure rate is relatively small. In the case of the TT method, the larger the timeout value, the larger the impact of the task failure rate. The reason for this is the difference in the task failure detections in the TT and SS methods. When a task fails, the TT method has to wait for the timeout in order to allocate a new replica to another host. Therefore, the large timeout value delays the start of the replica and also delays the task completion. On the other hand, a scheduler using the SS method can detect task failures immediately because the refresh messages stop when a task fails.

Fig. 3 shows the number of wasted CPU cycles. The impact of task failures in the TT method is almost the same regardless of the timeout value. This is because the replica creation in the TT method is triggered by a timeout and therefore is independent of task failures. On the other hand, the number of wasted CPU cycles in the SS method is almost proportional to the task failure rate. This implies that the number of replicas in the SS method increases with the number of task failures. However, these replicas reduce the job completion time of the SS method and therefore they are accepted as a necessary cost.

Fig. 4 and Fig. 5 show the performances of the two methods when $\mu$ is small. The impact of task failures is very small in both the TT and SS methods. The reason for this is that small tasks are not likely to be involved in host crashes because of their short stay in the hosts and they do not waste many CPU cycles even if they fail.

*2) Effect of the task timeout value:* It should be noted that the performance of the TT method is significantly affected by the task timeout value. Since the timeout value defines the maximum task execution time, the appropriate timeout value depends on the average task execution time. Calculation of the average task execution time requires information such as the average task size and the processing power of hosts. The fact that the performance of the TT method depends on this information contradicts the advantage of WQR, which does not require any information about tasks and hosts.

Moreover, even if we can calculate the average task execution time (denoted as $e$), the appropriate timeout value for the TT method cannot be determined by a simple calculation such as $e \cdot F$, where $F$ is some fixed value. There are two ways in which the task timeout value affects task execution.

1) If the timeout value is extremely large, the creation of a new replica of a failed task is delayed, and as a result, the completion of the task is also delayed.
2) If the timeout value is very small, many redundant replicas are created and they consume computing resources.

Fig. 2 (when $e$ is large) shows that a timeout value smaller than $e$ is desirable. However, Fig. 4 (when $e$ is small) shows the opposite behavior. These results indicate that effect (1) is dominant for large $e$ and effect (2) is dominant for small $e$. It is very difficult to estimate the effect of the timeout value on task execution for a given value of $e$.

*3) Effect of task cancellation:* It should be noted that in Fig. 3, the number of wasted CPU cycles in TT increases as the task failure rate decreases. The scheduler using TT is not able to cancel the execution of task replicas even if they are unnecessary. Therefore, even for a low task failure rate, many redundant replicas are executed to the end, and they waste many CPU cycles. On the other hand, the number of wasted CPU cycles in the SS method decreases as the task failure rate decreases. The scheduler using the SS method cancels task replicas as soon as they became unnecessary. Therefore, redundant replicas are cancelled before they waste CPU cycles. This result shows that SS's explicit cancellation of redundant replicas is effective in reducing the wastage of CPU cycles.

When the task failure rate is high, however, the difference between SS and TT is small. This is because a high task failure rate implies that many replicas including redundant ones fail frequently. Fig. 6 shows the wasted CPU cycles of failed tasks. From the graph, it can be observed that most of the wasted CPU cycles are those of failed tasks. In this case, most redundant replicas in SS failed before they were cancelled explicitly.

*4) Effect of message losses:* We also investigated the effect of changing the message loss rate of the simulations. The average task size is 10000 and the task failure rate is fixed to 1 (time/1000 s).

Fig. 7 and Fig. 8 show the job completion time and the number of wasted CPU cycles, respectively. It is obvious that the performance of the TT method is not affected by the message loss rate because the TT method does not exchange any messages during task execution. The performance of SS is also not affected as long as the message loss rate is lower than a threshold value (approximately 0.3 in the graph). However, as the message loss rate increases above the threshold, the performance of the SS method degrades significantly.

In the SS method, consecutive message losses cause false task abortion (as mentioned in section V-B). If the host aborts its task replica, the scheduler regards the task as failed and allocates a new task replica. However, when the message loss rate is very high, this new task replica may be aborted again. If the message loss rate is higher than the threshold, such abortions occur repeatedly, and the task completion is significantly delayed. The threshold is determined by the refresh interval and the timeout value of the SS method.

It should be noted that this result does not necessarily mean that the SS method is not robust against message losses. A significant degradation in performance occurs only when the high message loss rate continues permanently, which is impractical in real networks. A temporary increase in message losses does not affect the performance of SS.

## VII. Conclusions

In this paper, we investigated task scheduling in distributed computing. We selected WQR as the scheduling method because it does not require any kind of information about the hosts or tasks. We used the conventional task timeout method and the soft state method with WQR and compared their performances through simulations. The simulation results are summarized as follows.

- The soft state method is more robust than the task timeout method against task failures.
- The performance of the task timeout method depends on the timeout value, and it is difficult to calculate the appropriate timeout value.
- The soft state method wastes less CPU cycles than the task timeout method when the task failure rate is low.
- There is a threshold for the message loss rate, over which the performance of the soft state method degrades significantly.

From these results, we can conclude that the soft state method is preferable to the task timeout method for task scheduling in distributed computing.

## References

[1] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home: An Experiment in Public-Resource Computing," *Communications of the ACM*, vol. 45, no. 11, pp. 56–61, Nov. 2002.

[2] distributed.net, "Node Zero," http://www.distributed.net/.

[3] F. Dong and S. G. Akl, "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems," No.2006-504, Queen's University School of Computing, Tech. Rep., Jan. 2006.

[4] D. Paranhos, W. Cirne, and F. V. Brasileiro, "Trading Cycles for Information: Using Replication to Schedule Bag-of-Tasks Applications on Computational Grids," in *Proc. of the EuroPar 2003: Intl. Conf. on Parallel and Distributed Computing*, Klagenfurt, Austria, 2003, pp. 169–180.

[5] D. P. Anderson, "BOINC: A System for Public-Resource Computing and Storage," in *Proc. of 5th IEEE/ACM Intl. Workshop on Grid Computing*, Pittsburgh, PA, USA, 2004, pp. 4–10.

[6] W. Cirne, D. Paranhos, F. Brasileiro, and L. F. W. Goes, "On the Efficacy, Efficiency and Emergent Behavior of Task Replication in Large Distributed Systems," *Parallel Computing*, vol. 33, no. 3, pp. 213–234, 2007.

[7] D. P. Anderson, E. Korpela, and R. Walton, "High-Performance Task Distribution for Volunteer Computing," in *Proc. of First IEEE Intl. Conf. on e-Science and Grid Technologies*, Melbourne, Australia, 2005, pp. 196–203.

[8] J. C. S. Lui, V. Misra, and D. Rubenstein, "On the Robustness of Soft State Protocols," in *Proc. of 12th IEEE Intl. Conf. on Network Protocols (ICNP'04)*, Berlin, Germany, 2004, pp. 50–60.

[9] P. Ji, Z. Ge, J. Kurose, and D. Towsley, "A Comparison of hard-state and soft-state Signaling Protocols," in *Proc. of the 2003 Conf. on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe, Germany, 2003, pp. 251–262.

[10] D. Menascé, D. Saha, S. da S. Porto, V. Almeida, and S. Tripathi, "Static and Dynamic Processor Scheduling Disciplines in Heterogeneous Parallel Architectures," *Journal of Parallel and Distributed Computing*, vol. 28, pp. 1–18, 1995.

[11] H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman, "Heuristics for Scheduling Parameter Sweep Applications in Grid Environments," in *Proc. of the 9th Heterogeneous Computing Workshop (HCW'00)*, Cancun, Mexico, 2000, pp. 349–363.

[12] S. Raman and S. McCanne, "A Model, Analysis, and Protocol Framework for Soft State-based Communication," in *Proc. of ACM SIGCOMM*, Cambridge, MA, USA, 1999, pp. 15–25.

# First International Symposium on Multimedia—Applications and Processing

RECENT advances in pervasive computers, networks, telecommunications, and information technology, along with the proliferation of multimedia mobile devices —such as laptops, iPods, personal digital assistants (PDA), and cellular telephones—have stimulated the development of intelligent pervasive multimedia applications. These key technologies are creating a multimedia revolution that will have significant impact across a wide spectrum of consumer, business, healthcare, and governmental domains. Yet many challenges remain, especially when it comes to efficiently indexing, mining, querying, searching, and retrieving multimedia data.

The Multimedia—Processing and Applications 2008 (MMAP'08) Symposium addresses several themes related to theory and practice within multimedia domain. The enormous interest in multimedia from many activity areas (medicine, entertainment, education) led researchers and industry to make a continuous effort to create new, innovative multimedia algorithms and application.

As a result the conference goal is to bring together researchers, engineers and practitioners in order to communicate their newest and original contributions on topics that have been identified (see below). We are also interested in looking at service architectures, protocols, and standards for multimedia communications—including middleware—along with the related security issues, such as secure multimedia information sharing. Finally, we encourage submissions describing work on novel applications that exploit the unique set of advantages offered by multimedia computing techniques, including home-networked entertainment and games. However, innovative contributions that don't fit into these areas will also be considered because they might be of benefit to conference attendees.

Topics of interest are related to Multimedia Processing and Applications including, but are not limited to the following areas:

- Image and Video Processing
- Speech, Audio and Music Processing
- 3D and Stereo Imaging
- Distributed Multimedia Systems
- Multimedia Databases, Indexing, Recognition and Retrieval
- Data Mining
- E-Learning, E-Commerce and E-Society Applications
- Multimedia in Medical Applications
- Authentication and Watermarking
- Entertainment and Games
- Multimedia Interfaces

### General Chair

**Dumitru Dan Burdescu,** University of Craiova, Romania

### Steering Committee

**Dumitru Dan Burdescu,** University of Craiova, Romania
**Ioannis Pitas,** University of Thessaloniki, Greece
**Costin Badica,** University of Craiova, Romania
**Harald Kosch,** University Passau, Germany
**Vladimir Uskov,** Bradley University, USA
**Thomas M. Deserno,** Aachen University, Germany
**Mohammad S. Obaidat,** Monmouth University, USA

### Organizing Committee

**Dumitru Dan Burdescu,** University of Craiova, Romania
**Costin Badica,** University of Craiova, Romania
**Liana Stanescu,** University of Craiova, Romania
**Marius Brezovan,** University of Craiova, Romania

### Publicity Chairs

**Amelia Badica,** University of Craiova, Romania
**Dacheng Tao,** Hong Kong Polytechnic University, Hong Kong

### Program Committee

**Michael Auer,** Carinthia University of Applied Sciences, Austria
**Christopher Barry,** National University of Ireland, Ireland
**Laszlo Böszörmenyi,** Klagenfurt University, Austria
**Mohamad Bouhlel,** Sfax University, Tunisia
**David Bustard,** University of Ulster, UK
**Richard Chbeir,** Bourgogne University, France
**Ryszard Choras,** Institute of Telecommunications, Poland
**Vladimir Cretu,** Politehnica University of Timisoara, Romania
**Qi Chun,** Xi'an Jiaotong University, P.R.China
**Jean Louis Ferrier,** Angers University, France
**Rami Finkler,** Afeka College of Engineering, Tel Aviv, Israel
**Vladimir Fomichov,** State University Moscow, Russia
**Mislav Grgic,** University of Zagreb, Croatia
**Romulus Grigoras,** National Polytechnic Institute of Toulouse, France
**Daniel Grosu,** Wayne State University, USA
**Janis Grundspenkis,** Riga Technical University, Latvia
**Marek Hołyński,** Polish Information Processing Society, Poland
**Rajkumar Kannan,** Bishop Heber College, India
**Valery Korzhik,** State University of Telecommunications, St-Petersburg, Russia

**Vladik Kreinovich,** University of Texas El Paso, USA

**Igor Kotenko,** St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Russia

**Reggie Kwan,** Caritas Francis Hsu College, Hong Kong

**Michael Lang,** National University of Ireland, Ireland

**Rynson Lau,** Durham University, UK

**Ioan Alfred Letia,** Technical University of Cluj-Napoca, Romania

**Shiguo Lian,** France Telecom R&D Beijing, China

**Bogdan Logofatu,** University of Bucuresti, Romania

**Giuseppe Mangioni,** University of Catania, Italy

**Mihai Mocanu,** University of Craiova, Romania

**Marek Ogiela,** AGH University of Science and Technology, Poland

**Günther Pernul,** University of Regensburg, Germany

**Roland Rieke,** Fraunhofer-Institut Informationstechnologie, Darmstadt, Germany

**Boris Shishkov,** University of Twente, Netherlands

**Dacheng Tao,** Hong Kong Polytechnic University, Hong Kong

**Stefan Trausan-Matu,** Politehnica University of Bucharest, Romania

**Stefan Trzcielinski,** Poznan University of Technology, Poland

**George Tsihrintzis,** University of Piraeus, Greece

**Miguel Angel Vega-Rodríguez,** University of Extremadura, Spain

**Gregory Wojtkowski,** Boise State University, USA

**Wita Wojtkowski,** Boise State University, USA

**Franz Wotawa,** Technische Universitaet Graz, Austria

**Jacek Zurada,** University of Louisville, USA

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Similar Neighborhood Criterion for Edge Detection in Noisy and Noise-Free Images

Ali Awad and Hong Man
Stevens Institute of Technology Hoboken, NJ 07033 USA
Email: {aawad, hman}@stevens.edu}

***Abstract* — A novel approach for edge detection in noise-free and noisy images is presented in this paper. The proposed method is based on the number of similar pixels that each pixel in the image may have amongst its neighboring in the filtering window and within a pre-defined intensity range. Simulation results show that the new detector performs well in noise-free images but superior in corrupted images by salt and pepper impulse noise. Moreover, it is time efficient .**

## I. Introduction

THERE are many techniques in the literature used for edge detection some of them are based on error minimization [1], maximizing an object function [2], fuzzy logic [3], genetic algorithms [4], neural network[5], and Bayesian approach[6]. But the most popular approaches are the gradient- based filters such as Sobel filter [7], and Canny method [8]. However, they show unsatisfactory performance in noisy images. In this paper, we present a new method based on the similarity criteria by which any pixel in the image has a specific number of similar pixels in the filtering window and within a predefined intensity range is labeled as an edge point. In this approach, we say that pixel $y$ is similar to pixel $x$ if the absolute intensity difference between $x$ and y is ≤ $D$. Where $D$ is a pre-defined intensity value represents the maximum intensity difference between any two similar pixels. It is clear that, the edge pixel has a large intensity differences with its neighboring pixels [9]. Therefore, for a pre-defined value of $D$ we find that the similar pixels number of an edge pixel is small compared to that of a pixel located in a smoothing area. Thus, we can say that the location of each pixel in the image is specified by two factors. 1- The intensity difference of the pixel with its neighbors. 2- The number $N$ of similar pixel that any pixel in the image may have within the intensity range of [0, $D$]. As a result, we divide the pixel location into two sectors the first one includes the edge pixels and the second one contains the smoothing areas pixels. The general characteristics of the pixels in the first division are 1- They are very small number compared to the total number of the pixels in the image. 2- The intensity difference between the edge pixel and its surrounding ones is high 3-The similar number of the edge pixel is small. The pixels in the second division have the following features 1-They are majority, since they represent most of the image pixels. 2-The intensity differences between them are very small due to the homogeneity among them. 3-The numbers of their similar pixels are high. Let us look at the following example for pixels in 7x7 window from Lena image in Fig.1-a, we find that there are $N$=14 similar pixels make with the middle edge pixel $x$=218 intensity differences ≤ 20. $N$ is a small number because $x$ is located between two regions of high intensity variations, but in fig.1-(b), there are $N$=48 similar pixels make with $x$=194 intensity differences ≤ 20. $N$ in this case is a large number because $x$ is located among smoothing area pixels. That also is very obvious in Fig.2. It shows that 67.3% of all the Lena image pixels have similar pixels number between (45-48] and the other 32.7% remaining pixels have similar pixels number less than or equal to 45 pixels in a 7x7 window.

| | | | | | | |
|---|---|---|---|---|---|---|
| 240 | 227 | 207 | 182 | 169 | 159 | 159 |
| 247 | 237 | 217 | 192 | 175 | 158 | 156 |
| 249 | 239 | 229 | 205 | 185 | 164 | 158 |
| 251 | 243 | 232 | 218 | 198 | 175 | 165 |
| 255 | 247 | 240 | 232 | 213 | 189 | 173 |
| 255 | 252 | 244 | 235 | 221 | 200 | 182 |
| 255 | 252 | 249 | 239 | 226 | 211 | 193 |

(a)

| | | | | | | |
|---|---|---|---|---|---|---|
| 193 | 187 | 187 | 195 | 195 | 194 | 200 |
| 192 | 187 | 186 | 194 | 195 | 194 | 199 |
| 191 | 189 | 186 | 194 | 194 | 194 | 199 |
| 191 | 190 | 186 | 194 | 195 | 194 | 199 |
| 190 | 190 | 187 | 196 | 196 | 195 | 201 |
| 189 | 191 | 188 | 197 | 197 | 196 | 202 |
| 189 | 191 | 189 | 198 | 198 | 197 | 203 |

(b)

Fig 1. 7x7 windows in different regions in Lena image.(a) shows pixels in abrupt areas (b) pixels in smoothing areas.

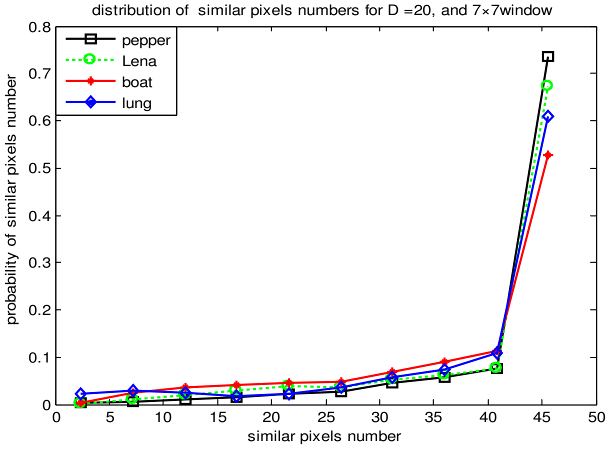


Fig.2 The distribution of similar pixels numbers within a 7×7 window size and within an intensity difference ≤ 20 for different images.

Fig.3- *a, b, c,* and *d* show the locations of the pixels that have similar pixels number $N$ ≤ 40 within $D$=20, $N$ ≤ 25 within $D$=20, $N$ ≤ 10 within $D$=10 and $N$ ≤ 5 within $D$=5, respectively for Lena image. It is obvious that all of those pixels are minority pixels and located on the edges of the image. Therefore, we can define a new parameter called similarity parameter $S = \dfrac{D}{N}$ to measure the similarity of a

978-83-60810-14-9/08/$25.00 © 2008 IEEE

pixel with its surrounding ones in the filtering window within an intensity range [0, *D*] and would give an acceptable edge detection results. It is clear from the last example that at $S \approx 1$ satisfactory performance is obtained. This value can be used as a threshold.

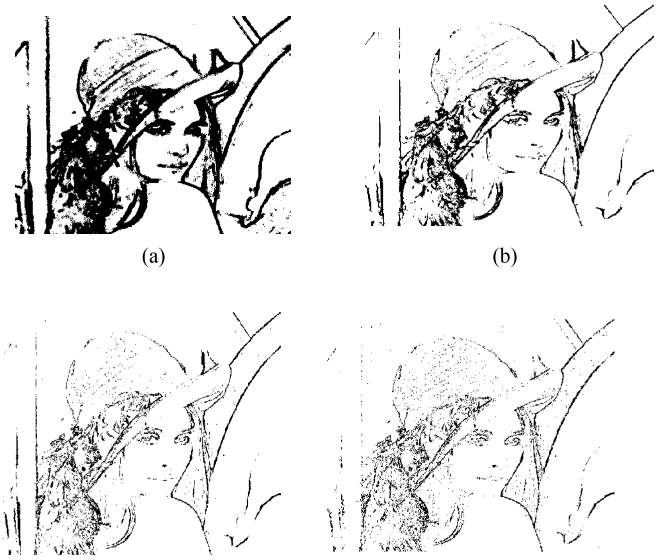


Fig.3 The locations of the minority pixels in Lena image based on their similar pixels number *N within* an intensity range [0, *D* ] (a) $N \leq 40$, $D$ =20 (b) $N \leq 25$, $D$ =20, (c) $N \leq 10$, $D$ =10, (d) $N \leq 5$, $D$ =5.

## II. Algorithm Description

Define the filtering window $W(p_{ij})$ of $k \times k$ size centered at the pixel $p_{i,j}$ and the location $(i, j)$ in the image *P*. In this algorithm each pixel $p_{ij}$ is transformed to a binary value $L_{ij}$ based on a pre-defined value of $S^{th}$ in each phase *r* as :

$$T(p_{ij,r}) = L^r{}_{ij} \quad \forall p_{ij,r} \in P_r \tag{1}$$

$$L^r_{ij} = \begin{cases} 255 & if \;\; S_r \, {}^{\geq}\!/\!_{\leq} \, S_r{}^{th} \\ 0 & else \end{cases} \tag{2}$$

$$S_r = \frac{D_r}{N_r} \leq \frac{D_r}{k^2 - 1} \qquad r = [1,2] \tag{3}$$

${}^{\geq}\!/\!_{\leq}$ means one of the two signs $\{\geq \; or \leq\}$ . As *S* decreases the similarity level increases and hence more edge pixels detected. $S^{th}$ is the threshold of the similarity level that helps judging if the current pixel *x* is an edge pixel or not. The proposed approach consists of two phases as illustrated below:

### *A. Edge Identification Phase*

In this phase we try to differentiate between the edge pixels and the other pixels in the image as the following. Take the absolute intensity difference between the current central pixel $p_{i,j}$ and each of its surrounding pixels in the window as:

$$d(p_{i-s,j-t}, p_{i,j}) = |p_{i-s,j-t} - p_{i,j}| \tag{4}$$

where $\{s, t = 0 \pm 1, \ldots, \pm \frac{k-1}{2}, (s,t) \neq (0,0)\}$

Then, count the number $C(p_{ij})$ of the pixels in the filtering window that make intensity differences $\leq D$ with $p_{i,j}$

$$C(p_{\,ij}) = \; number \; \left\{ d(p_{i-s,j-t}, p_{i,j}) \leq D \right\} \tag{5}$$

any pixel $p_{i\text{-}s,j\text{-}t}$ satisfies (5) is called a similar pixel to $p_{ij}$. The pixel $p_{ij}$ is considered as an edge point and replaced by a zero in the corresponding location in another binary image *U* if $C(p_{ij}) \leq S_1^{th}$ , otherwise it is replaced by 255 as:

$$U(u_{i,j}) = \begin{cases} 0, & C(p_{i,j}) \leq S_1^{th} \\ 255 & else \end{cases} \tag{6}$$

Note that, the threshold $S_1^{th} \cong 1$ delivers satisfactory results for many images.

### *B. Complementary phase:*

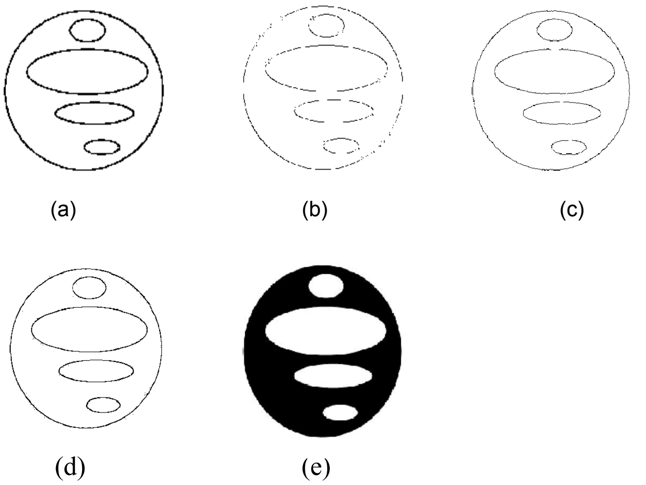
If the detected image is noise –free image, then we have to stop by phase 1. But if the detected image is a noisy image then image *U* will contain all the detected edge pixels, the noisy pixels that satisfy the first threshold, and the background white pixels. Note that the noisy pixels in image *U* have a small number of similar pixels compared to that of the edge pixels, because the edge pixels are condensed along the edge lines; however the line shape, but the noisy pixels are scattered through the background pixels (white pixels) which make a large intensity differences with the noisy pixels. Also, we can increase the similar pixels number for the edge pixels by increasing the first threshold. Therefore, for extracting the edge points we need to repeat step 1 and 2 on the image *U.* Any pixel $u_{ij} = 0$ in *U* is considered as an edge point in a binary image *V* if $C(u_{ij}) \geq$ the second similarity parameter $S_2^{th}$ , otherwise it is replaced by 255 in *V a*s the following:

$$V(v_{i,j}) = \begin{cases} 0, & C(u_{ij}) \geq S_2^{th} \\ 255 & else \end{cases} \tag{7}$$

$S_2^{th}$ should be decreased as the noise rate increases and vice versa. In all the simulation experiments we maintain the value of *D* constant and change only the value of *N* .

## III. Simulation Results

To show the performance of the proposed approach we apply it on different 8-bit grey-level images and compare the results with other well known methods in the literature as Sobel and Canny detectors. The results are compared subjectively in terms of the edge quality, and computationally in terms of the relative processing time in seconds for the different methods (measured by MATLAB COMAND “etime”). CPU of 1.73GHZ and RAM of 1MB are used in all the simulation experiments.7×7 window size is used with the proposed detector. Similarity parameter $S_1^{th} \cong 0.8$ is used with the noise-free image and in the 1st phase of the noisy images. The threshold *Th*=120 is used with Sobel method and the ones that are used with Canny approach are $Th_{max}$=120, $Th_{min}$=70.

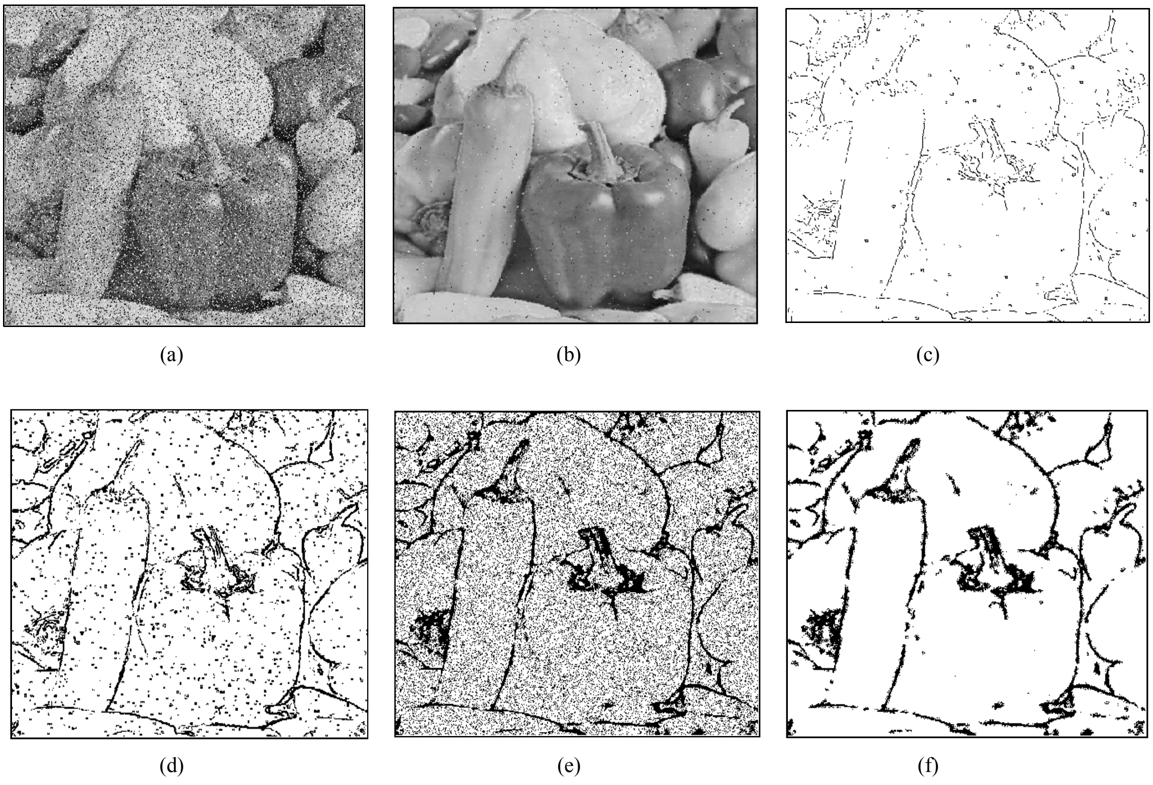

(a) (b) (c)

(d) (e)

Fig.4 Edge detection results for different filters on a synthetic image: (a) Sobel (t= **2.5 sec** ), *Th* =120 (b) Canny (t= **5.1 sec** ), $Th_{max}$=120, $Th_{min}$=70(c) EMO (t= **25.1 sec**), (d) New (t= **3.9** sec), ($D_l$=20, $N_l$=25), (e) original image.

Fig. 4 is applied on a noise-free synthetic image of 323×393 size. In this experiment we compare our method with EMO approach [2] that is proposed for uncorrupted images, besides Sobel and Canny methods. It is clear that the proposed method shows a continuous thin edge line while the other methods suffer either from discontinuity in the edge line or show a thicker edge. Besides, the processing time of the proposed detector is comparable to Canny and Sobel filters while it is faster 6.4 times than the EMO filter, see the caption of fig. 4. The slow convergence that shown by the EMO method is due to the enormous using of the subtraction operations, i.e., it needs 4x18 subtraction operations in each window in the four directions.

Fig. 5 and Fig. 6 show the results of different methods for edge detection in corrupted pepper and Lena images with 20% and 25% salt and pepper impulse noise rates, respectively. Pepper and Lena images are both of 512× 512 sizes. For appropriate comparison the corrupted images are firstly restored by using 3 ×3 median filter since it high efficiency in impulse noise removal, and then we apply the Sobel and Canny approaches on the restored images. It is noticeable that the proposed method delivers better performance than Canny, and Sobel methods since it effectively removes the impulse noise and maintains the main image features, while the other methods are still contain residual noise and miss some of the image details. The edge lines that are obtained by the new method look somewhat thick; the reason is that some of the neighboring noisy pixels are able to satisfy the threshold criterion. However, they illustrate the main feature of the image to be used for any higher level image processing task. Moreover, the proposed detector is faster than the Sobel and Canny approaches, respectively. Note that the computational times


(a) (b) (c)

(d) (e) (f)

Fig. 5 Edge detection for 20% impulse noise corrupted pepper image: (a) corrupted image, (b) restored version by median filter,(c) Canny after restored image, time= **21.3** sec, $Th_{max}$ =120, $Th_{min}$ =70 (d) Sobel after restored image, time= **15.1** sec , *Th=120* (e) proposed filter after corrupted image -1 st phase ( $N_1^{th}$ =25, $D_1$ =20) time=7 sec (f) proposed filter-2 nd phase ( $N_2^{th}$ = 20, $D_2$ =20,), time= **12.3** sec

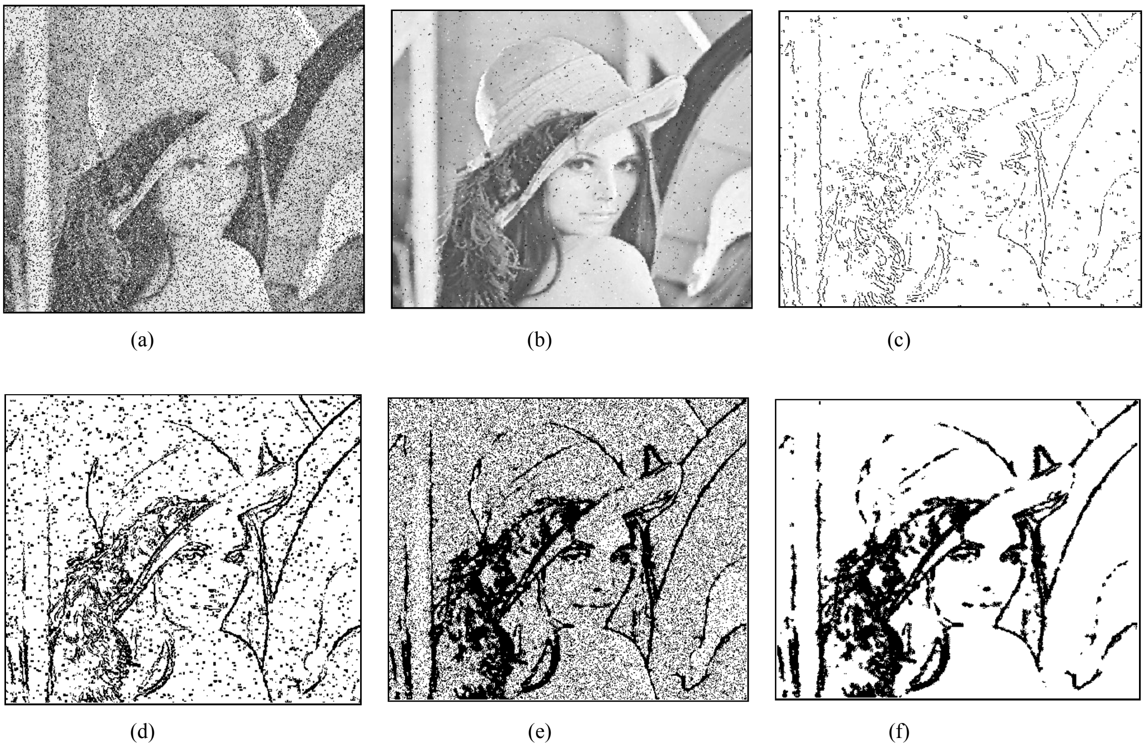


Fig.6 Edge detection for 25% impulse noise corrupted Lena image: (a) corrupted image, (b) restored version by median filter,(c) Canny after restored image, time= **21.3** sec, $Th_{max}$ =120, $Th_{min}$ =70 (d) Sobel after restored image , time= **15.1** sec, $Th$= 120 (e) proposed filter -1 st phase after corrupted image( $N_1^{th}$ =25, $D_1$ =20) time=7 sec (f) proposed filter-2 nd phase ( $N_2^{th}$ = 25, $D_2$ =20,), time= **12.3** sec,

that are obtained after one or more iterations are the same for all the methods. In the 2nd phase the edge quality increases as the similarity parameter decreases and vice versa. The reason is that, as the noise rate increases the number of similar pixels for the residual noise in the 1st phase images increases. Therefore, we have to decreases $S_2^{th}$ by increasing $N_2$ as shown in fig. 5 and 6. Since the edge pixels have larger number of similar pixels than any other noisy pixels, we expect that most of the original edge pixels will satisfy the second similarity parameter threshold.

## IV. Conclusion

A high performance edge detection approach based on the similarity criteria is proposed in this paper. In which, the pixels that have minimum numbers of similar pixels is consider as edge pixels in free-noise images, and the pixels that have maximum numbers of similar pixels are considered as edge pixels in the noisy images. Simulations results indicate that the proposed approach achieves superior performance than other well known methods, particularly in images corrupted by impulse noise. Moreover, it is time efficient method and has a low computational complexity.

## References

[1] C. Spinu, Garbay, and J. M. Chassery Edge detection by estimation and minimization of errors," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 695–704, August 1997.

[2] C. C. Kang, and W. J. Wang, "A novel edge detection method based on the maximizing objective function," *Pattern Recognition*, vol. 40 no.2,pp. 609–618, Feb. 2007.

[3] J. Wu, Z. Yin, and Y. Xiong, "The fast multilevel fuzzy edge detection of blurry images," *IEEE Signal Processing Letters,* vol. 4, no. 5, pp. 344–347, May 200.

[4] L. Caponetti, N. Abbattists, and G. Carapella, "A genetic approac to edge detection," in *Int. Conf. Image Processing*, vol. 94, 1994, pp. 318–322.

[5] V. Srinivasan, P. Bhatia, and S. H. Ong, "Edge detection using neural network," *Pattern Recognit.*, vol. 27, no. 12, pp. 1653–1662, 1995.

[6] T. J. Hebert and D. Malagre, "Edge detection using a priori model," in *Int. Conf. Image Processing*, vol. 94, 1994, pp. 303–307.

[7] R. Gonzalez, and R. Wood, "Digital Image Processing," Addison Wesley, 1992, pp 414–428.

[8] J. Canny, "A computational approach to edge detection," *IEEE Tranns. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–697, June 1986.

[9] E. Abdou and W. K. Pratt, "Quantitative design and evaluation enhancement/thresholding edge detectors, " Proc. IEEE , vol. 67, pp. 753–763, 1979.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Accurate Localization in Short Distance based on Computer Vision for Mobile Sensors

Kwansik Cho, Ha Yoon Song, Jun Park
Department of Computer Engineering, Hongik University, Seoul, Korea
mstr50@gmail.com, hayoon@wow.hongik.ac.kr, jpark@cs.hongik.ac.kr

***Abstract*—In order to maximize the utilization of mobile sensor network, formation of sensor set and localization of each sensor node must be implemented. It is required that localization is one of the most important functionality of mobile sensor nodes. In this paper, we present a technique which improves the relative location of the MSN with a computer vision technology.**

**This technique effects only in short distance but only with low price sensors, we achieved precise localization in the resolution of 10 centimeters. The well known perspective-3-point problem have been exploited for the precise short distance localization. By experiment we present an interrelation between angle of camera view and a LED pattern interval. We measures the distance of the counterpart vehicle and vehicles shares distance information of obstacle and the relative vehicles with possible cooperation of vehicles. The angle of a vehicle can be identified by digital compass. Finally, with a share of location information we can achieve localization of mobile sensor nodes with high accuracy.**

## I. Introduction

THIS research is regarding short distance localization with use of computer vision technology under Mobile Sensor Network (MSN) environment. In order to maximize the utilization of mobile sensor network, formation of sensor set and localization of each sensor node must be implemented.

We built several Mobile Sensor Vehicles (MSVs) as mobile sensor nodes for a mobile sensor network. Each MSV can identify locations of companion MSV and can share such information in order to achieve accurate localization of whole MSVs. We used a micro camera as a visual sensor of MSV. The result of localization can be presented on maps based on grid or topology. We believe this research can be applied formation of UAV (Unmanned aerial vehicle), a formation for a deep-sea fishing vessel and other related area.

This paper is structured as follows. In section II we will show related problems and possible solutions of localization. Section III discusses retailed experiment procedure for localization and section IV will show the result of localization. The final section V will conclude this paper with discussions of possible future research.

## II. Research Background

There have been variety forms of Mobile Sensor Nodes (MSN) which utilizes various techniques of localizations such as RSSI, GPS, Raider, Laser, Camera, and so on [1] [2]. One of the most prominent one, an RSSI based localization, usually measures radio signal strength and it works well with popular network devices. Moreover, an 802.11 device based software approach can be realize easily. However, RSSI method is prone to be fragile with a presence of obstacles or so which will diminish or attenuate radio signal strength. In a short distance, RSSI signals usually is too high that nullify accurate localization thus it is good for long distance, low accurate localization.

For medium distance, a trajectory based approach is a useful one. Usually mobility is recorded and the accumulated records are used to calculate current location of mobile sensor nodes. However, also traveling errors are accumulated as traveling distance increased.

In this paper we will discuss about short distance, high accurate localization. It is based on computer vision and of course it has limitations of distance with visibility.

We combined these three levels of localization technique and will focus on computer vision based one. For the purpose of experiments, we designed and implemented Mobile Sensor Vehicle (MSV) which has all three levels of localization features aforementioned as shown in [3].

In this section, we will discuss about related problem.

### A. Identification of Colleague MSVs as a base for localization

Localization is required for MSN in order to maximize its usefulness. However, a single MSV cannot locate its location precisely. The ultimate localization can only be done with the cooperation of nodes in MSN.

The first requirement for localization is to identify the location of colleague MSV as a base point. For this purpose, we prepared three facilities for each MSV. Each MSV can estimate its location by trajectory trail. Moreover, each MSV can identify other colleague MSV with their infrared LED signal. In addition, this location information can be communicated by wireless network device equipped with each MSV.

Of course, a camera or a set of cameras are installed on an MSV in order to identify colleague MSVs. This set of cameras have infrared filters in order to diminish the effect of extra light noise in operating environment.

### B. Location Determination Problem

With a set of camera, the required information for localization is collected from the view of cameras. For example, an infrared LED light can be a parameter to calculate the colleague's location. In this research, we applied two previous results. The first one is Sample Consensus(RANSAC) Method [4] and the second one is PnP Method [5] [6].

This work was supported by 2007 Hongik University Research Fund.

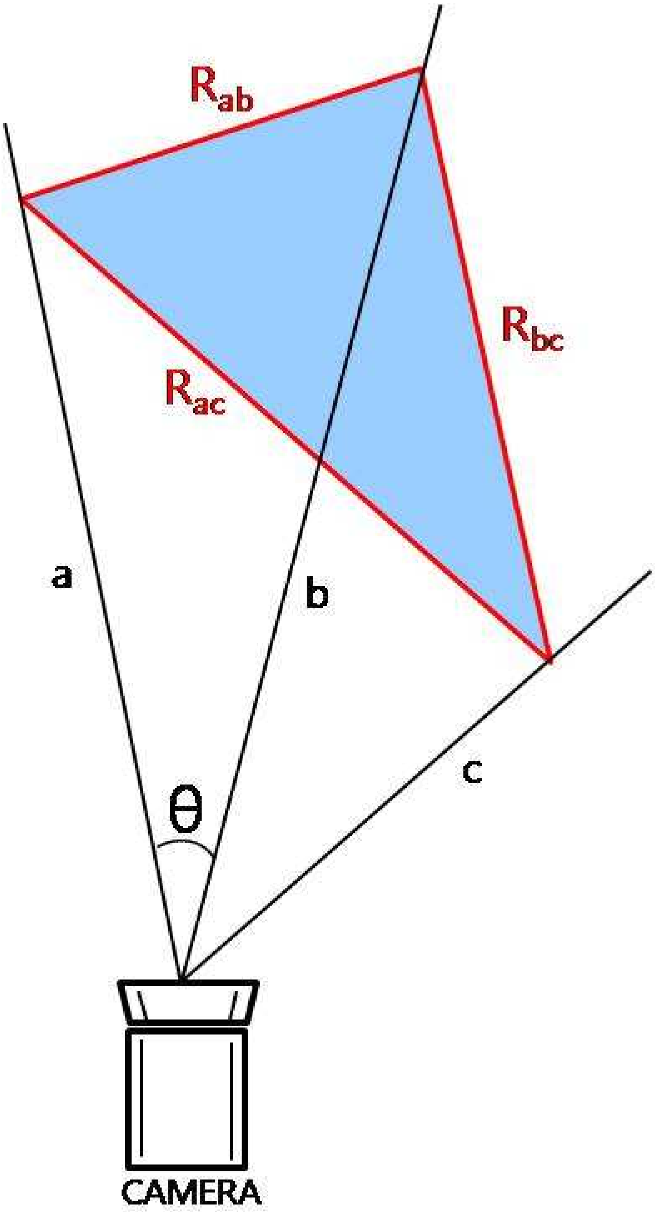


Fig. 1. Calculation of distance from Camera Lens to Vertex of triangle ABC

For RANSAC method, because of least square method, there is no possibility of wrong computation with gross error value. This is the major reason why we applied RANSAC method. In order to solve the problem of converting 3-dimensional view to 2-dimensional camera image, which has lost distance problem, we applied perspective-3-point (P3P) problem.

Figure 1 shows the basic principle of P3P problem. The gray triangle is composed by infrared LED installed on each MSV. Points $A, B, C$ stand for each infrared LEDs and these vertices compose a triangle. The distance $R_{ab}, R_{bc}, R_{ac}$ is known constants.

From figure 1 we can drive the following, very well-known, mathematical equation as shown in equation 1.

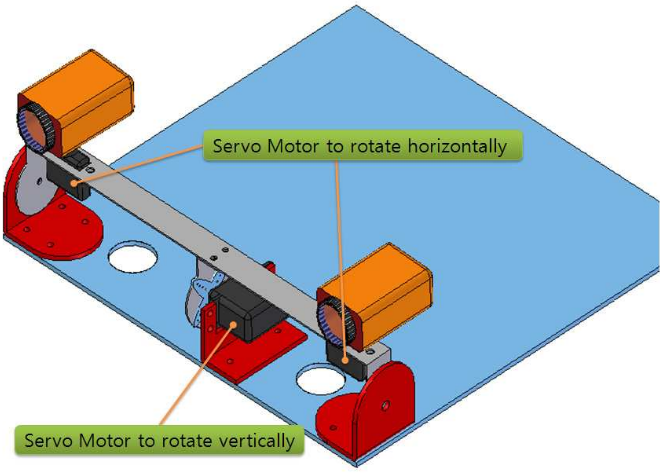

Servo Motor to rotate horizontally
Servo Motor to rotate vertically

Fig. 2. Stereo Eye System

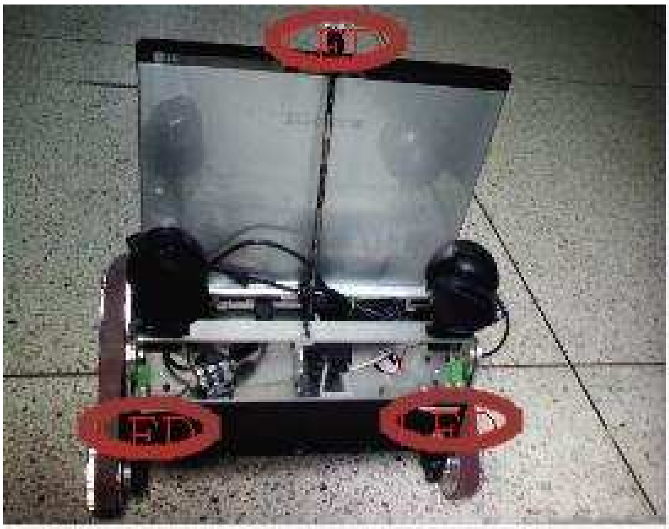
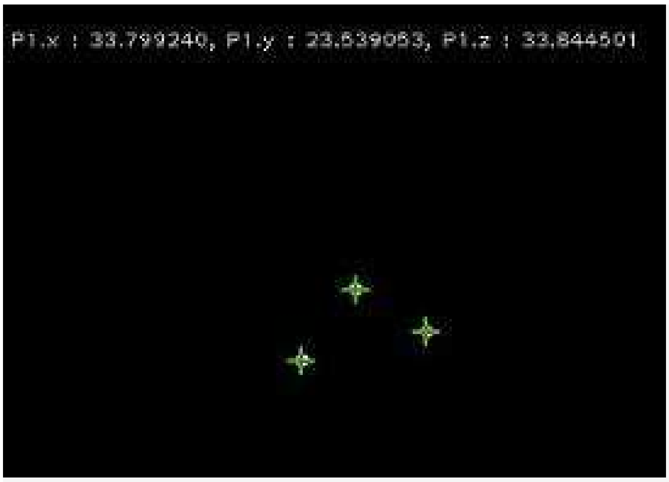


Fig. 3. Mobile Sensor Vehicle for Experiment

$$\begin{aligned} {R_{ab}}^2 &= a^2 + b^2 - 2ab \cdot cos\theta_{ab} \\ {R_{ac}}^2 &= a^2 + c^2 - 2ac \cdot cos\theta_{ac} \\ {R_{bc}}^2 &= b^2 + c^2 - 2bc \cdot cos\theta_{bc} \end{aligned} \tag{1}$$

The equation 1 is in closed form. The number of solutions from these equations will be up to eight. However, there are up to four positive roots.

With this P3P based method, we can only measure distances between observer and observed. For precise localization, we must identify angle of MSVs as well. Our MSV is equipped with digital compass in order to identify the angle of MSV based on magnetic poles. As predicted, digital compass also has its own error in angle measurement but is tolerable.



P1.x : 33.799240, P1.y : 23.539053, P1.z : 33.844501

Fig. 4. Tracking Program View

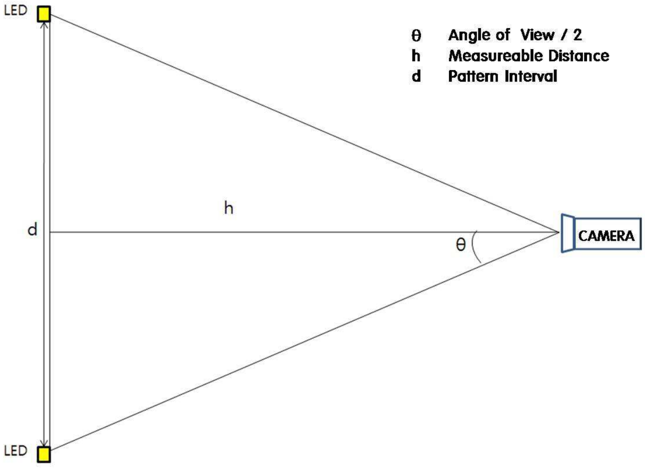


Fig. 5. Correlation of LED pattern interval and measurable minimum distance

TABLE I
IR-LED SPECIFICATIONS

| MODEL NO. | Half Angle | Peak Wavelegnth |
|---|---|---|
| SI5315-H | ±30° | 950nm |
| OPE5685 | ±22° | 850nm |
| OPE5194WK | ±10° | 940nm |
| TLN201 | ±7° | 880nm |
| EL-1KL5 | ±5° | 940nm |

## III. Implementation and Experiments

### A. Experimental Environment

Each MSV has a set of Infrared LED (IR-LED) in a form of triangle and the length of edges are all 30 centimeters. The IR lights from these LEDs can be viewed by stereo camera system from other colleague MSV. The stereo eye system as shown in figure 2 has two cameras. Three servo motors controls two stereo eyes vertically and horizontally.

The stereo cameras are equipped with IR filters. The front view of MSV for these equipments are as shown in figure 3. Three IR-LEDs forms a triangle and a stereo camera system are also presented.

With fixed length of triangle edges, i.e. interval between IR-LED, is fixed by 30 centimeters. Therefore by using P3P method, the distance between camera and MSVs with IR-LED triangle can be calculated. Embedded software for each MSV has a realtime part for P3P solution. The software also shows the image from stereo camera as a part of P3P solution as shown in figure 4.

The ideal situation starts by estimating the angle between two cameras. Once camera direction is fixed, we can estimate angles between tracked object and cameras, however, MSV can move every direction which causes difficulties to measure that angle. Moreover, if these cameras have pan-tilt functionalities, it is impossible to measure such an angle in real time.

Another method with P3P technique is to assume the distance to the object. The distance to object and the scale of triangle in camera view is proportional inversely thus the size of LED triangle can be a starting point to estimate the distance to obstacles. We decided to standardize the reduced scale of LED triangle in order to estimate distance to objects. The basic concept of this method is depicted in figure 5. We will discuss this figure in the subsection III-C in detail.

This approach has limits of camera visibility, i.e. objects beyond visibility cannot be identified. However, two other localization methods are already prepared for beyond sight localization as described in II. In addition with the help of digital compass, we can measure the direction of each MSV. The combination of this information can achieve short distance accuracy for localization.

### B. Preliminary Experiment

We conduct preliminary experiment in order to choose optimal device for computer vision based localization. The first purpose of this experiment is to select the best LED in order to increase the range of localization. Our past result showed 250 centimeter of localization range however our aim is to enlarge the range to 400 centimeters or farther.

We choose five infrared light emitting diodes with typical characteristics. We first concentrated on the visible angle of LED lights since we assumed wider visible angle guarantees the clearer identification of LED light and more precise localization.

Table I shows the specifications of various IR-LEDs with visible angle and peak wavelength. The major reason why we choose those IR-LEDs are as follows:

- Smaller half angle of LED enables long distance tracking however increase invisibility from the side.
- Larger half angle of LED enables tracking from the side however decrease tracking distance.

With infrared filter equipped cameras we planned experiments to evaluate the LEDs for vision based localization. Table II show the result of visible distance and visibility of IR-LEDs. Twelve experiments have been made and average

TABLE II
IR-LED VISIBILITY EXPERIMENTS

| MODEL NO. | Max Length | Visible Angle | Visibility |
|---|---|---|---|
| SI5315-H | 500cm | ±60° | Stable |
| OPE5685 | 490cm | ±45° | Somewhat Unstable |
| OPE5194WK | 520cm | ±35° | Most Stable |
| TLN201 | 510cm | ±20° | Unstable |
| EL-1KL5 | 450cm | ±10° | Indiscriminable |

values are shown. From the specifications of IR-LEDs, 5 volts DC voltage is supplied for the experiment.

Among five IR-LEDs, two showed stable visibility and acceptable visibility distance. Between these two candidates, we finally choose the best LED of MODEL NO.SI5313-H since it has the widest half angle as well with reasonable visibility distance.

### C. Main Experiments

Figure 5 shows the relationship between LED triangle size ($d$) and distance from camera to LED triangle ($h$). The relation between $d$ and $h$ can be directly drawn from the following equation 2

$$\begin{aligned} \tan\theta &= \frac{d}{2h} \\ \theta &= \arctan\frac{d}{2h} \end{aligned} \tag{2}$$

Most of cameras has angle of view in $54^\circ \sim 60^\circ$. Since we used camera with angle of view in $60^\circ$, from the equation 2 we can solve ratio about $h : d = 1 : 1.08$. The actual value of $d$ is 30 centimeter for our experiment.

Thus we can summarize the following:

- High angle of view camera can increase minimum measure distance.
- With narrow LED pattern interval, we can decrease actual distance $h$ but practically meaningless.
- With wider LED pattern interval, we can increase actual distance but dependent on MSV size.

From the experiments, we can identify the vision based localization is effective within the range from 30 centimeters to 520 centimeters with our LogiTech CAM camera. The 30 centimeter lower bound is due to the 30 centimeter interval of LED triangle edges. The 520 centimeter upper bound is due to the visible sight capability of LogiTech CAM camera. Thus 520 centimeter would be a maximum distance of computer vision based localization. However it is still meaningful since we can achieve very high accuracy in localization with these cheap, low grade cameras. The other idea for more localization distance is to use cameras with higher resolution.

From our experiments, we identified the correlation between actual distance from camera to colleague MSV and size of

TABLE III
CORRELATION OF ACTUAL DISTANCE ($h$) AND RELATIVE SIZE OF LED TRIANGLE

| A Regular Triangle (30cm) LED Pattern | |
|---|---|
| Distance (Cm) | Relative Size Calculated by P3P |
| 70 | 95.20465 |
| 80 | 84.94255 |
| 90 | 76.69515 |
| 100 | 70.98945 |
| 110 | 66.14175 |
| 120 | 63.67415 |
| 130 | 60.56855 |
| 140 | 57.90765 |
| 150 | 54.40745 |
| 160 | 51.12315 |
| 170 | 49.34895 |
| 180 | 47.84185 |
| 190 | 45.20355 |
| 200 | 43.00485 |
| 210 | 41.95535 |
| 220 | 39.15445 |
| 230 | 37.31475 |
| 240 | 36.35485 |
| 250 | 35.09785 |
| 260 | 34.46155 |
| 270 | 33.87925 |
| 280 | 32.94845 |
| 290 | 31.01745 |
| 300 | 30.20645 |
| 310 | 29.77155 |
| 320 | 28.12365 |
| 330 | 27.65485 |
| 340 | 26.37745 |
| 350 | 25.54655 |
| 360 | 24.78645 |
| 370 | 24.57855 |
| 380 | 23.54155 |
| 390 | 22.78955 |
| 400 | 22.57515 |
| 410 | 22.54655 |
| 420 | 21.78945 |
| 430 | 21.89485 |
| 440 | 20.79875 |
| 450 | 20.42085 |
| 460 | 19.88265 |
| 470 | 19.07455 |
| 480 | 18.78955 |
| 490 | 18.77985 |
| 500 | 17.57515 |
| 510 | 17.54655 |
| 520 | 16.82315 |

LED triangles in camera view. Table III shows correlations between actual distance and triangle size in camera view. The results in table III can be translated into graphical form as shown in figure 6.

From figure 6 the result shows the fluctuation of results with more than 430 centimeters which makes localization unstable. For applications which require the error range of 20 centimeters, we can use the results to 520 centimeters. Since our aim is to keep localization errors within the range of 10 centimeters, we decided to discard results more than 430 centimeters.

## IV. Experimental Result

From our experiment in the previous subsections we will provide the final result of computer vision based localization in this subsection.

Table IV shows the final result. Figure 7 shows graphical version of table IV.

Apart from the results in previous section, these table and figure shows actual distance up to 420 centimeters. From figure 6 we can observe errors in calculated values of P3P for more than 420 centimeter distance. These errors is due to the resolution limit of CAM camera which is $640 \times 480$. Even a small noise can vary actual distance of ten centimeters in the distance more than 420 centimeters.

Thus we conclude the accurate localization by computer vision can be done in the range of 70 centimeters to 420 centimeters with our camera equipments.

For the localization in more than 420 centimeters, localization based on MSV trajectory tracking will be effective. In addition, for the localization in more than 30 meters, localization based on RSSI will be effective [3].

## V. Conclusion and Future Research

We built mobile sensor vehicle as nodes for mobile sensor network. Our mobile sensor nodes has capabilities in multilevel localization. In this paper we discussed regarding computer vision based localization.

We can achieve very precise and short distance localization with the help of computer vision technology.

Within the distance of 420 centimeters, we can identify the location of each mobile sensor nodes in a resolution of 10 centimeters. And within the distance of 520 centimeters, we still have the possibility of localization in a resolution of 20 centimeters.

However our experiment has several limitations due to the resolution of cameras and the vertices distance of LED triangle. Since the vertices distance of LED triangle is limited by the chassis size of MSV, we cannot extend the distance more than current configuration.

Therefore for future researches, we plan to upgrade cameras and the angle of view for cameras as well for longer localization distance. With this better sensors and calibration of sensors we expect that we can extend the localization distance up to 1,000 centimeters.

TABLE IV
Calculated distance from measured relative distance

| Relative Distance Measured | Actual Distance Assumed (Cm) |
|---|---|
| 99.99999 ∼ 95.20465 | 70 |
| 95.20464 ∼ 84.94255 | 80 |
| 84.94254 ∼ 76.69515 | 90 |
| 76.69514 ∼ 70.98945 | 100 |
| 70.98944 ∼ 66.14175 | 110 |
| 66.14174 ∼ 63.67415 | 120 |
| 63.67414 ∼ 60.56855 | 130 |
| 60.56854 ∼ 57.90765 | 140 |
| 57.90764 ∼ 54.40745 | 150 |
| 54.40744 ∼ 51.12315 | 160 |
| 51.12314 ∼ 49.34895 | 170 |
| 49.34894 ∼ 47.84185 | 180 |
| 47.84184 ∼ 45.20355 | 190 |
| 45.20354 ∼ 43.00485 | 200 |
| 43.00484 ∼ 41.95535 | 210 |
| 41.95534 ∼ 39.15445 | 220 |
| 39.15444 ∼ 37.31475 | 230 |
| 37.31474 ∼ 36.35485 | 240 |
| 36.35484 ∼ 35.09785 | 250 |
| 35.09784 ∼ 34.46155 | 260 |
| 34.46154 ∼ 33.87925 | 270 |
| 33.87924 ∼ 32.94845 | 280 |
| 32.94844 ∼ 31.01745 | 290 |
| 31.01744 ∼ 30.20645 | 300 |
| 30.20644 ∼ 29.77155 | 310 |
| 29.77154 ∼ 28.12365 | 320 |
| 28.12364 ∼ 27.65485 | 330 |
| 27.65484 ∼ 26.37745 | 340 |
| 26.37744 ∼ 25.54655 | 350 |
| 25.54654 ∼ 24.78645 | 360 |
| 24.78644 ∼ 24.57855 | 370 |
| 24.57854 ∼ 23.54155 | 380 |
| 23.54154 ∼ 22.78955 | 390 |
| 22.78954 ∼ 22.57515 | 400 |
| 22.57514 ∼ 22.54655 | 410 |
| 22.54654 ∼ 21.78945 | 420 |

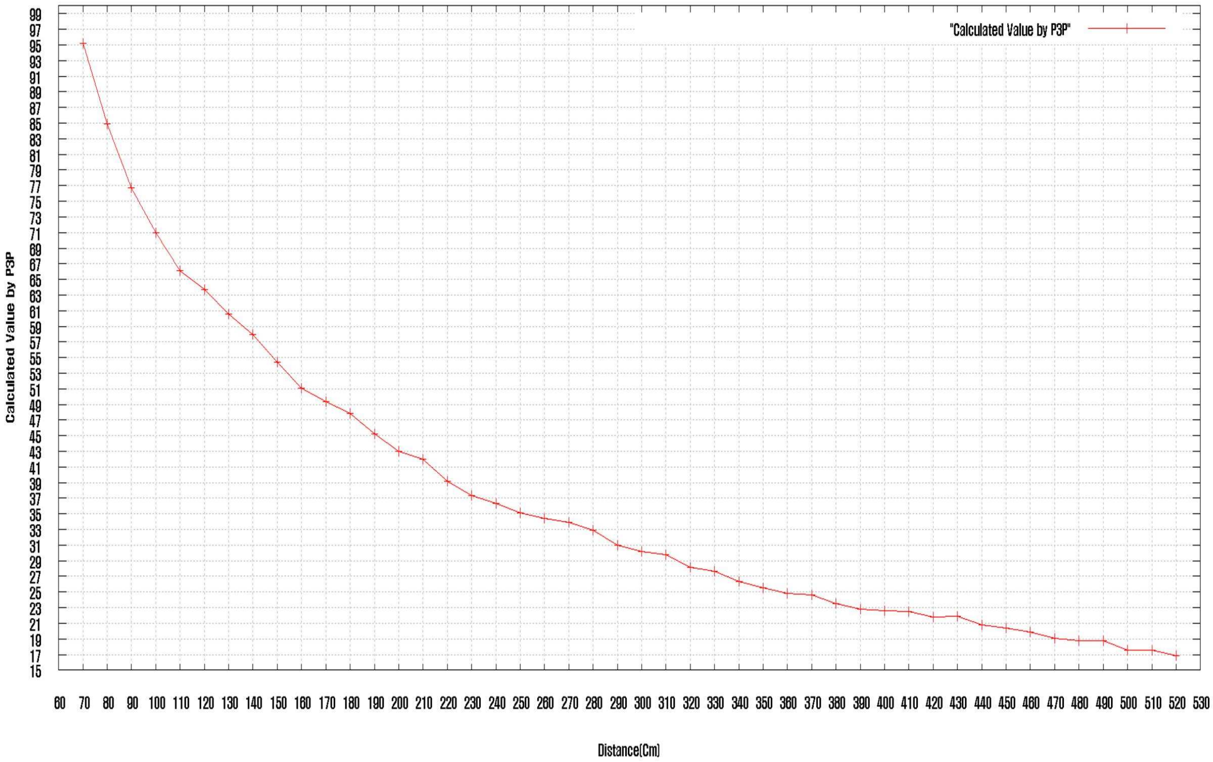


Fig. 6. Relative size of triangle calculated by P3P on actual distance

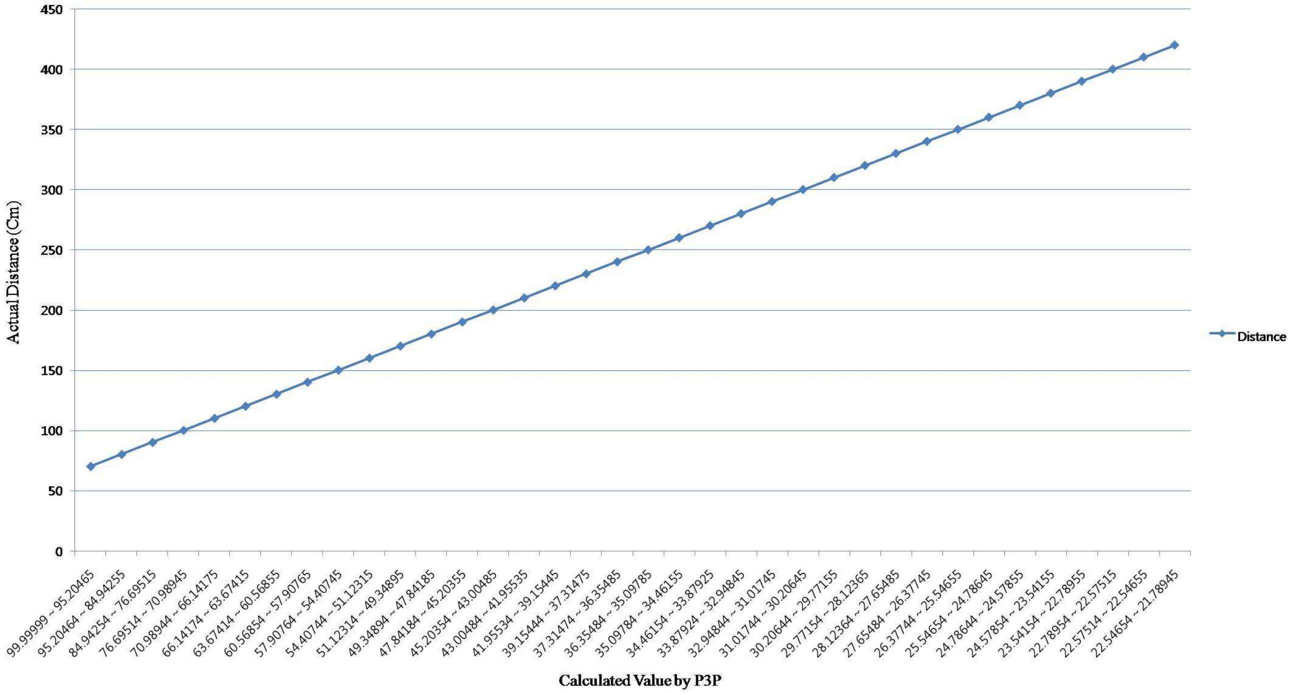


Fig. 7. Actual distance calculated from measured relative distance

We believe we showed an example of high accuracy localization and this research will help researchers for such applications in a field of mobile sensor network as well as robotics.

With these precise localization methods, a sensor formation technique will be a feasible one.

## REFERENCES

[1] J.-S. Gutmann, W. Burgard, D. Fox, K. Konolige, "An Experimental Comparion of Localization Method," IROS, 1998.

[2] J.-S. Gutmann, C. Schlege, AMOS, "Comparison of Scan Matching approaches for self-Localization in indoor Environments," EUROBOT, 1996.

[3] Jae Young Park, Ha Yoon Song, "Multilevel Localization for Mobile Sensor Network Platforms," International Workshop on Real Time Software (RTS'08), *International Multiconference on Computer Science and Information Technology, 20–22 October 2008, Wisla, Poland,* IEEE Press.

[4] M. A. Fischler, R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. of the ACM, Vol. 24, pp. 381–395, 1981.

[5] Xiao Shan Gao and Xiao Roing Hou and Jianliang Tang and Hang?Fei Cheng, "Complete Solution Classification for the Perspective Three Point Problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, August 2003.

[6] Haralick, R.M and Lee, D and Ottenburg, K and Nolle, M, "Analysis and Solutions of The Three Point Perspective Pose Estimation Problem," IEEE Computer it Vision and Pattern Recognition Computer Society Conference on, 592-598, June 1991.

[7] David Nister, "A Minimal Solution to the Generalized 3-Point Pose Problem," IEEE CVPR, 2004.

[8] Dimitrios Karagiannis, Alessandro Astolfi, "A New Solution to the Problem of Range Identification in Perspective Vision Systems," *IEEE Transactions on Automatic Control,* vol. 50, NO. 12, December 2005.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 495 – 500

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Character Recognition Using Radon Transformation and Principal Component Analysis in Postal Applications

Mirosław Miciak
University Technology and Life Sciences in Bydgoszcz
Faculty of Telecommunications and Electrical Engineering
ul. Kaliskiego 7, 85-791 Bydgoszcz, Poland
Email: miciak@utp.edu.pl

***Abstract*—This paper describes the method of handwritten characters recognition and the experiments carried out with it. The characters used in our experiments are numeric characters used in post code of mail pieces. The article contains basic image processing of the character and calculation of characteristic features, on basis of which it will be recognized. The main objective of this article is to use Radon Transform and Principal Component Analysis methods to obtain a set of features which are invariant under translation, rotation, and scaling. Sources of errors as well as possible improvement of classification results will be discussed.**

## I. Introduction

THE today's systems of automatic sorting of the post mails use the OCR (Optical Character Recognition) mechanisms. In the present recognizing of addresses (particularly written by hand) the OCR is insufficient.

The typical system of sorting consists of the image acquisition unit, video coding unit and OCR unit. The image acquisition unit sends the mail piece image to the OCR for interpretation. If the OCR unit is able to provide the sort of information required (this technology has 50 percent effectiveness for all mails), it sends this data to the sorting system, otherwise the image of the mail pieces is sent to the video coding unit, where the operator writes down the information about mail pieces.

The main problem is that operators of the video coding unit have lower throughput than an OCR and induce higher costs [1]. Therefore the OCR module is improving, particularly in the field of recognition of the characters. Although, these satisfactory results were received for printed writing, the handwriting is still difficult to recognize. Taking into consideration the fact, that manually described mail pieces make 30 percent of the whole mainstream, it is important to improve the possibility of segment recognizing the hand writing. This paper presents the proposal of a system for recognition of handwritten characters, for reading post code from mail pieces.

The process of character recognition process can be divided into stages: filtration and binaryzation, normalization, Radon transform calculating, accumulator analysis, Principal Component Analysis, feature vector building, and character recognition stage.

The first step of the image processing is binarization. The colourful image represented by 3 coefficients Red, Green and Blue from the acquisition unit must be converted to the image with 256 levels of grey scale. The next step of processing of the image of mail piece is digital filtration. The filtration is used for improving the quality of the image, emphasizing details and making processing of the image easier. The filtration of digital images is obtained by

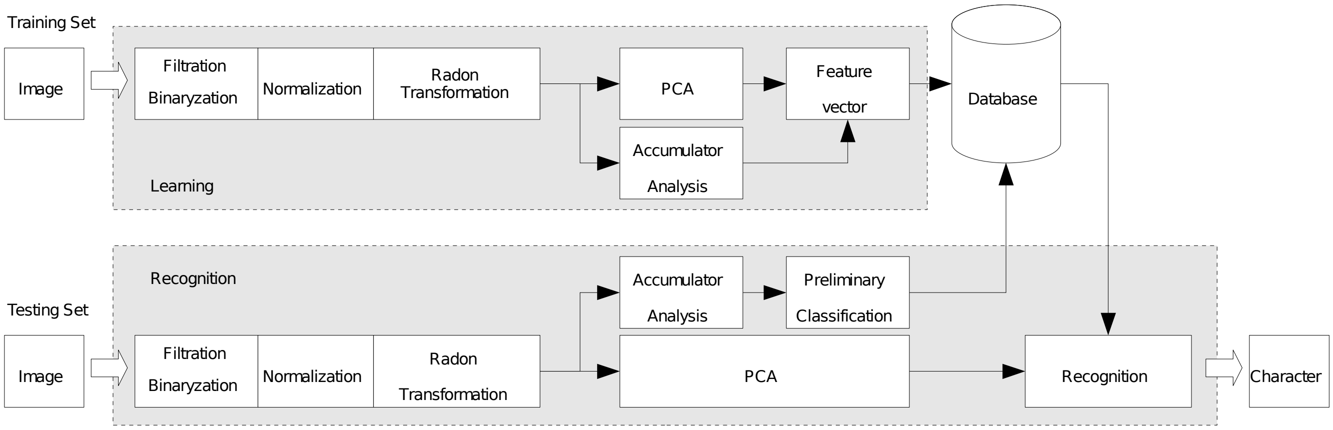
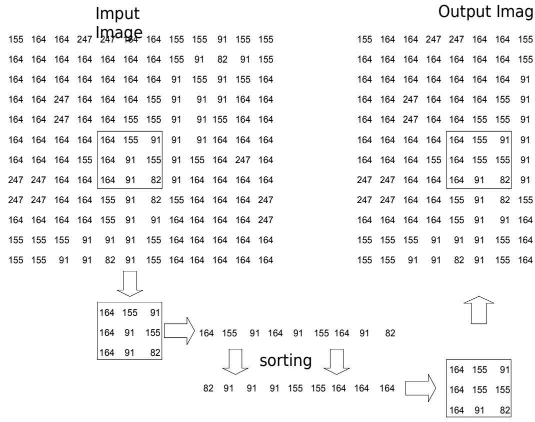


Fig 1. Character recognition system

978-83-60810-14-9/08/$25.00 © 2008 IEEE

convolution operation. The new value of point of image is counted on the basis of neighbouring points value. Every value is classified and it has influence on new value of point of the image after filtration [2].



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 155 | 164 | 164 | 247 | 247 | 164 | 164 | 155 | 155 | 91 | 155 | 155 |
| 164 | 164 | 164 | 164 | 164 | 164 | 164 | 155 | 91 | 82 | 91 | 155 |
| 164 | 164 | 164 | 164 | 164 | 164 | 164 | 91 | 155 | 91 | 155 | 164 |
| 164 | 164 | 247 | 164 | 164 | 164 | 155 | 91 | 91 | 91 | 164 | 164 |
| 164 | 164 | 247 | 164 | 164 | 155 | 155 | 91 | 91 | 155 | 164 | 164 |
| 164 | 164 | 164 | 164 | 164 | 155 | 91 | 91 | 91 | 164 | 164 | 164 |
| 164 | 164 | 164 | 155 | 164 | 91 | 155 | 91 | 155 | 164 | 247 | 164 |
| 247 | 247 | 164 | 164 | 164 | 91 | 82 | 91 | 164 | 164 | 164 | 164 |
| 247 | 247 | 164 | 164 | 155 | 91 | 82 | 155 | 164 | 164 | 164 | 247 |
| 164 | 164 | 164 | 164 | 155 | 91 | 91 | 164 | 164 | 164 | 164 | 247 |
| 155 | 155 | 155 | 91 | 91 | 91 | 155 | 164 | 164 | 164 | 164 | 164 |
| 155 | 155 | 91 | 91 | 82 | 91 | 155 | 164 | 164 | 164 | 164 | 164 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 155 | 164 | 164 | 247 | 247 | 164 | 164 | 155 | 155 | 91 | 155 | 155 |
| 164 | 164 | 164 | 164 | 164 | 164 | 164 | 155 | 91 | 82 | 91 | 155 |
| 164 | 164 | 164 | 164 | 164 | 164 | 164 | 91 | 155 | 91 | 155 | 164 |
| 164 | 164 | 247 | 164 | 164 | 164 | 155 | 91 | 91 | 91 | 164 | 164 |
| 164 | 164 | 247 | 164 | 164 | 155 | 155 | 91 | 91 | 155 | 164 | 164 |
| 164 | 164 | 164 | 164 | 164 | 155 | 91 | 91 | 91 | 164 | 164 | 164 |
| 164 | 164 | 164 | 155 | 164 | 155 | 155 | 91 | 155 | 164 | 247 | 164 |
| 247 | 247 | 164 | 164 | 164 | 91 | 82 | 91 | 164 | 164 | 164 | 164 |
| 247 | 247 | 164 | 164 | 155 | 91 | 82 | 155 | 164 | 164 | 164 | 247 |
| 164 | 164 | 164 | 164 | 155 | 91 | 91 | 164 | 164 | 164 | 164 | 247 |
| 155 | 155 | 155 | 91 | 91 | 91 | 155 | 164 | 164 | 164 | 164 | 164 |
| 155 | 155 | 91 | 91 | 82 | 91 | 155 | 164 | 164 | 164 | 164 | 164 |

| | | |
|---|---|---|
| 164 | 155 | 91 |
| 164 | 91 | 155 |
| 164 | 91 | 82 |

164 155 91 164 91 155 164 91 82

82 91 91 91 155 155 164 164 164

| | | |
|---|---|---|
| 164 | 155 | 91 |
| 164 | 155 | 155 |
| 164 | 91 | 82 |

Fig 2. Median filtering

In the pre-processing part non-linear filtration was applied. The statistical filter separates the signal from the noise, but it does not destroy useful information. The applied filter is median filter, with mask *3*x*3*.

The image of character received from the acquisition stage have different distortion such as: translation, rotation and scaling. The character normalization is applied for standardization size of the character. Images there are translated, rotated and expanded or decreased. The typical solutions takes into consideration the normalization coefficients and calculate the new coordinates given by:

$$[x,y,1]=[i,j,1]\begin{bmatrix}1&0&0\\0&1&0\\-I&-J&1\end{bmatrix}\times \begin{bmatrix}m_i&0&0\\0&m_j&0\\0&0&1\end{bmatrix}\times\begin{bmatrix}\cos\beta&\sin\beta&0\\-\sin\beta&\cos\beta&0\\0&0&1\end{bmatrix} \qquad (1)$$

where: *I,J* is a center of gravity given by :

$$I=\frac{\sum_i\sum_j if(i,j)}{\sum_i\sum_j f(i,j)} \qquad J=\frac{\sum_i\sum_j jf(i,j)}{\sum_i\sum_j f(i,j)} \qquad (2)$$

In the reality we haven't got this parameters starting right now, so we use new coordinate system where center is equals to center of gravity of the character. The value of angle rotation is according to main axes of the image. The value of scale coefficient is calculated by mean value of variation of the character. So the center of gravity of the character is good candidate point of the center of image as a product of normalization stage.

## II. Radon Transformation

In recent years the Radon transform have received much attention. This transform is able to transform two dimensional images with lines into a domain of possible line parameters, where each line in the image will give a peak positioned at the corresponding line parameters. This have lead to many line detection applications within image processing, computer vision, and seismic [3][18]. The Radon Transformation is a fundamental tool which is used in various applications such as radar imaging, geophysical imaging, nondestructive testing and medical imaging [20].

The Radon transform computes projections of an image matrix along specified directions. A projection of a two-dimensional function *f(x,y)* is a set of line integrals. The Radon function computes the line integrals from multiple sources along parallel paths, or beams, in a certain direction. The beams are spaced 1 pixel unit apart. To represent an image, the radon function takes multiple, parallel-beam projections of the image from different angles by rotating the source around the centre of the image. The "Fig.3" shows a single projection at a specified rotation angle.

The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle. The radial coordinates are the values along the $x'$ -axis, which is oriented at $\theta$ degrees counter clockwise from the $x$ -axis. The origin of both axes is the center pixel of the image .

For example, the line integral of *f(x,y)* in the vertical direction is the projection of *f(x,y)* onto the $x$ -axis; the line integral in the horizontal direction is the projection of *f(x,y)* onto the $y$ -axis. The "Fig.4" shows horizontal and vertical projections for a simple two-dimensional function.

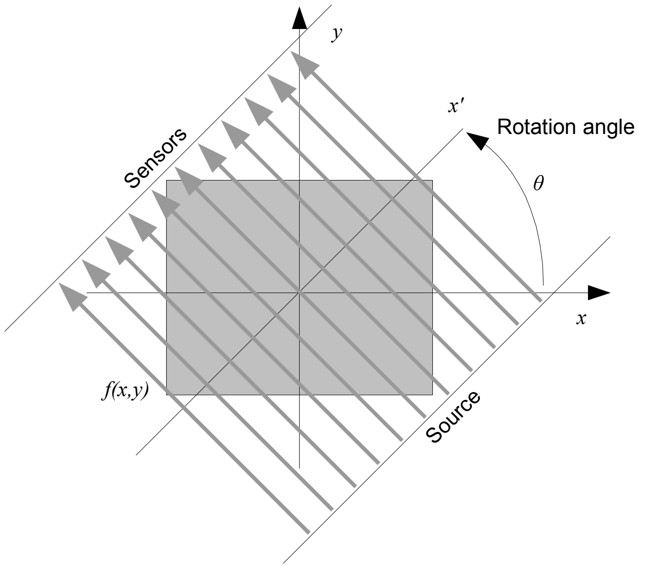


Fig 3. Single projection at a specified rotation angle

Projections can be computed along any angle $\theta$, by use general equation of the Radon transformation [23][24][25] :

$$R_{\Theta}(x')=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(xcos\Theta+ysin\Theta-x')\,dydy \qquad (3)$$

where $\delta\ (\cdot)$ is the *delta* function with value not equal zero only for argument equal 0, and:

$$x'=xcos\Theta+ysin\Theta \qquad (4)$$

$x'$ is the perpendicular distance of the beam from the origin and $\theta$ is the angle of incidence of the beams. The "Fig. 5" illustrates the geometry of the Radon Transformation. The

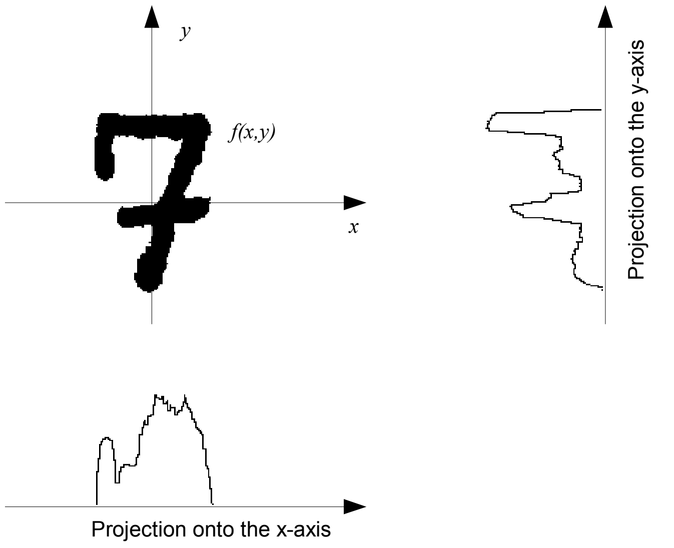


Fig 4. Horizontal and Vertical Projections of a Simple Function

very strong property of the Radon transform is the ability to extract lines (curves in general) from very noise images. Radon transform has some interesting properties relating to the application of affine transformations. We can compute the Radon transform of any translated, rotated or scaled image, knowing the Radon transform of the original image and the parameters of the affine transformation applied to it.

This is a very interesting property for symbol representation because it permits to distinguish between transformed objects, but we can also know if two objects are related by an affine transformation by analyzing their Radon transforms [19]. It is also possible to generalize the Radon transform in order to detect parametrized curves with non-linear behavior [3][4][5].

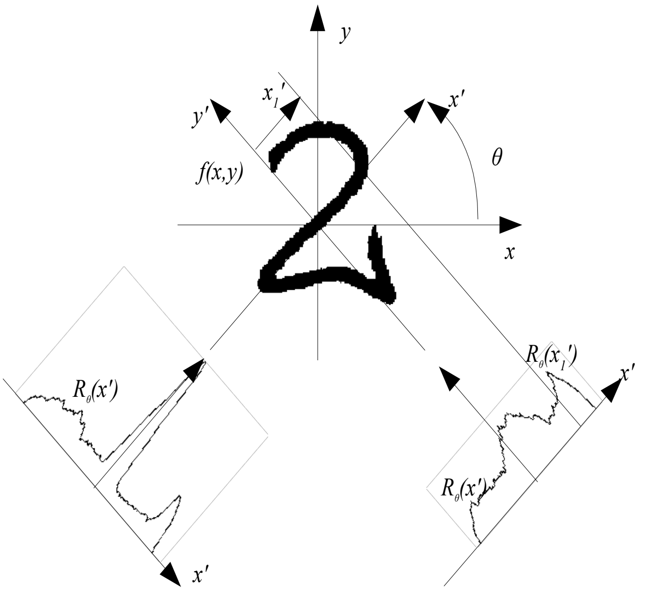


Fig 5. Geometry of the Radon Transform

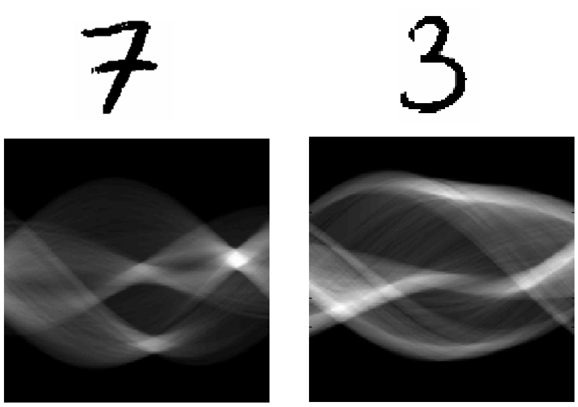


Fig 6 Sample of accumulator data of Radon Transformation

## III. Principal Components Analysis

PCA is mathematically defined [6][7][8] as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA is theoretically the optimum transform for a given data in least square terms. The main idea of using PCA for character recognition is to express the large 1-D vector of pixels constructed from 2-D character image into the compact principal components of the feature space [21].

PCA can be used for dimensionality reduction in a data set by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data.

For all $r$ digital images *f(x,y)* from the normalization stage is creating column vector $X_k$ by the concatenate operation, where $k=(1,...,r)$ . For that prepared images we can calculate mean of brightness intensity $M$, difference vector $R$ and covariance matrix $\Sigma$ .

$$M_k = \frac{1}{r} \sum_{k=1}^{r} X_k \qquad (5)$$

$$R_k = X_k - M_k \qquad (6)$$

$$\sum = \frac{1}{r} \sum_{k=1}^{r} R_k R_k^t \qquad (7)$$

where:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ . \\ X_r \end{bmatrix} \qquad (8)$$

$$M = \begin{bmatrix} M_1 \\ M_2 \\ . \\ M_r \end{bmatrix} \qquad (9)$$

$$R=\begin{bmatrix} R_1 \\ R_2 \\ . \\ R_r \end{bmatrix} \qquad (10)$$

Principal components are calculating from the eigenvectors $\Phi_l$ and eigenvalues $\lambda_l$ of the covariance matrix $\Sigma$. The eigenvectors $\Phi_l$ are normalized, sorted in order eigenvalue, highest to lowest and transponed, to obtain transformation matrix $W$, where $K$ is the number of dimensions in the dimensionally reduced subspace calculated by:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{l} \lambda_i} \geq p \qquad (11)$$

where: $p$ is assumed as threshold [21]. The matrix $W$ is given by:

$$W=\begin{bmatrix} \Phi_1^1 & \dots & \Phi_1^K \\ . & \dots & . \\ \Phi_l^1 & \dots & \Phi_l^K \end{bmatrix} \qquad (12)$$

After image projection into eigenvectors space we do not use all eigenvectors, but these with maximum eigenvalues, this gives the components in order of significance. The eigenvector associated with the largest eigenvalue is one that reflects the greatest variance in the input data. That is, the smallest eigenvalue is associated with the eigenvector that finds the least variance. They decrease in exponential fashion, meaning that the roughly 90% of the total variance is contained in the first 5% to 10% of the dimensions [21].

The projection of $X$ into eigenvectors space is given by:

$$Y=W(X-M) \qquad (13)$$

where:

$$Y=\begin{bmatrix} Y_1 \\ Y_2 \\ . \\ Y_r \end{bmatrix} \qquad (14)$$

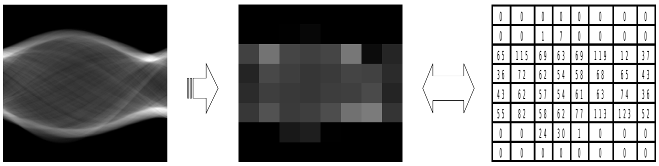
The final data set will have less dimensions than the original [8], after all we have $r$ column-vector for each input image with $K$ values *:*

$$Y_k=(y_{1,} y_2, \dots, y_K)^t \qquad (15)$$

The PCA module in proposal system generate a set of data, which can be used as a features in building feature vector section. For instance when we use input matrix 8x8 from Radon transformation stage, as a result we obtained $K$=8 values vector, using Cattell's criterion [9].

## IV. Feature Vector

Two sets of data received from the PCA module and accumulator analysis stage are used to create vector of features of character. The amount of data from Radon transformation depends from the image of character size and numbers of projections. For example, we use image with size 128x128 pixels and step $\theta$ equals one degree. With those parameters we can retrieve accumulator matrix with 180 width and 185 height cells. To produce feature vector we don't use all values from the accumulator. The reduction the accumulator data is possible by the resizing operation - when generally size of matrix is most commonly decreased. The most known scaling techniques are: method used with Pixel art scaling algorithms, Bicubic interpolation, Bilinear interpolation, Lanczos resampling, Spline interpolation, Seam carving [10][11][12]. In our research we make tests with Linear, Bicubic and Bilinear methods. The results with other methods was very similar and do not have influence on the recognition rate of proposed system. As a result of resize operation in our system is a matrix 8x8 elements as in "Fig.7".



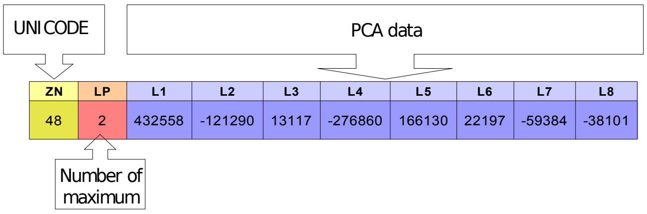
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 7 | 0 | 0 | 0 | 0 |
| 65 | 115 | 69 | 63 | 69 | 119 | 12 | 37 |
| 36 | 72 | 62 | 54 | 58 | 68 | 65 | 43 |
| 43 | 62 | 57 | 54 | 61 | 63 | 74 | 36 |
| 55 | 82 | 58 | 62 | 77 | 113 | 123 | 52 |
| 0 | 0 | 24 | 30 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fig 7. Scaling operation

The next step of vector features preparing is concatenate operation and Principal Component Analysis of resized matrix from Radon Transformation, see on "Fig.8".

Fig 8. Creating vector $X$ by concatenate operation.

As a result of PCA is 8 element vector *L1-L8* of main values from input data, which will be used to feature vector. The second set of data are: code of known character *ZN* as a Unicode [13] and number of local maximum *LP* from the Accumulator Analysis stage. The feature vector consists a 10 values "Fig.9".



| ZN | LP | L1 | L2 | L3 | L4 | L5 | L6 | L7 | L8 |
|---|---|---|---|---|---|---|---|---|---|
| 48 | 2 | 432558 | -121290 | 13117 | -276860 | 166130 | 22197 | -59384 | -38101 |

Fig 9. Creating feature vector in proposed system

## V. Preliminary Classification

The aim of the preliminary classification is to reduce the number of possible candidates for an unknown character, to a subset of the total character set. For this purpose, the selected domain is categorized into six groups with number of local maximum as in "Fig.10".

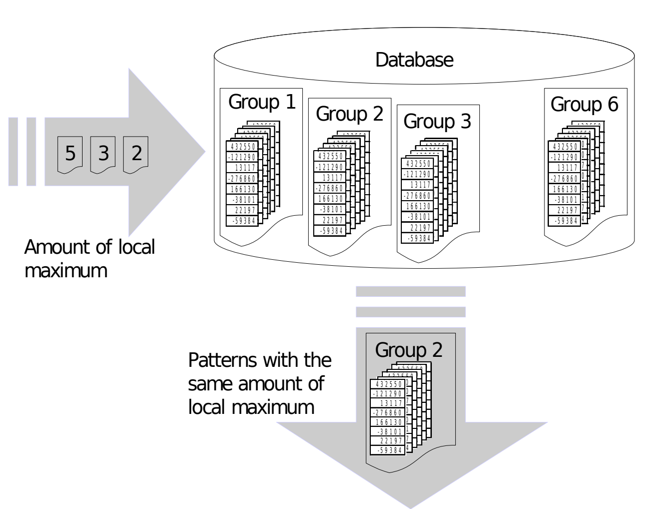


Fig 10. The organizations of feature vectors in database

The preliminary classification is based on the amount of local maximum calculating in the Accumulator Analysis stage "Fig.11".

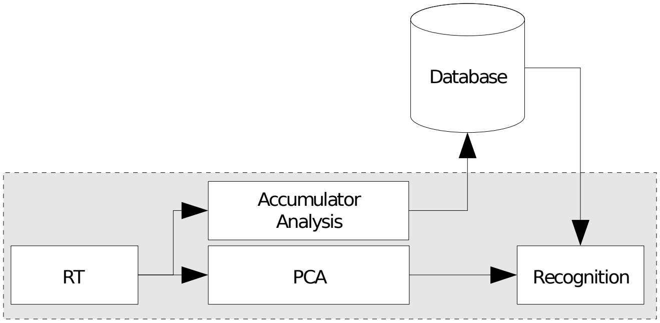


Fig 11. The Preliminary classification scheme

## VI. Recognition and Classification

The classification in the recognition module compared features from the pattern to model features sets obtained during the learning process. Based on the feature vector *Z* recognition, the classification attempts to identify the character based on the calculation of Euclidean distance [22] between the features of the character and of the character models [14].

The distance function is given by:

$$D(C_i, C_r)=\sum_{j=1}^{N}[R(j)-A(j)]^2 \qquad (16)$$

where:
*Ci* - is the predefined character,
*Cr* - is the character to be recognized,
*R* - is the feature vector of the character to be recognized,
*A* - is the feature vector of the predefined character,
*N* - is the number of features.

The minimum distance *D* between unknown character feature and predefined class of the characters is the criterion choice of the character [14].

## VII. The Experiment

For evaluation experiments, we extracted some digit data from various paper documents from different sources eg. mail pieces post code, bank cheque etc. In total, the training datasets contain the digit patterns of above 130 writers. Collected 920 different digits patterns for training set and 300 digits for test set. Each pattern is represented as a feature vector of 10 elements.

Comparing results for handwritten character with other researches is a difficult task because are differences in experimental methodology, experimental settings and handwriting database. Liu and Sako [15] presented a handwritten character recognition system with modified quadratic discriminant function, they recorded recognition rate of above 98 %. Kaufman and Bunke [16] employed Hidden Markov Models for digits recognition. They obtained a recognition rate of 87 %. Aissaoui and Haouari [14] using Normalized Fourier Descriptors for character recognition, obtained a recognition rate above 96 %. Bellili using the MLP-SVM recognize achieves a recognition rate 98 % for real mail zip code digits recognition task [17]. In this experiment recognition rate 94 % was obtained. The detailed results for individual testing sets was presented in Table I.

Table I.
Recognition rate for testing sets

| Testing Set | Recognition rate |
|---|---|
| Set 1 | 94.7 % |
| Set 2 | 94.2 % |
| Set 3 | 93.3 % |

## VIII. Conclusions

The selecting of the features for character recognition can be problematic. Moreover fact that the mail pieces have different sizes, shapes, layouts etc. this process is more complicated. The paper describes often used the character image processing such as image filtration, binaryzation, normalization and the Radon Transformation calculating.

The character recognition algorithms were proposed. In connection with this work the application included the algorithms is in progress. So far the application reached recognition speed 30 characters/sec without any optimization.

In the future work is planning to use another statistical methodology such as JDA/LDA. Moreover the will be upgraded to remaining all alphanumerical signs and special signs often placed on regular post mails.

## References

[1] G. Forella, "Word perfect", *Postal Technology*, UKIP Media & Events Ltd, UK 2000.

[2] J. Rumiński, "Metody reprezentacji, przetwarzania i analizy obrazów w medycynie",

[3] T. Peter, "The Radon Transform - Theory and Implementation", PhD thesis, Dept. of Mathematical Modelling Section for Digital Signal Processing of Technical University of Denmark, 1996.

[4] R. N. Bracewell, "Two-Dimensional Imaging", *Englewood Cliffs*, Prentice Hall, 1995, pp. 505-537.

[5] J. S. Lim, " Two-Dimensional Signal and Image Processing", *Englewood Cliffs*, Prentice Hall, 1990, pp. 42-45

[6] I. T. Jolliffe, "Principal Component Analysis", *Springer Series in Statistics*, 2nd ed., Springer, 2002.

[7] J. Shlens, "A Tutorial on Principal Component Analyzing", available at: http://www.cs.cmu.edu/~elaw/papers/pca.pdf.

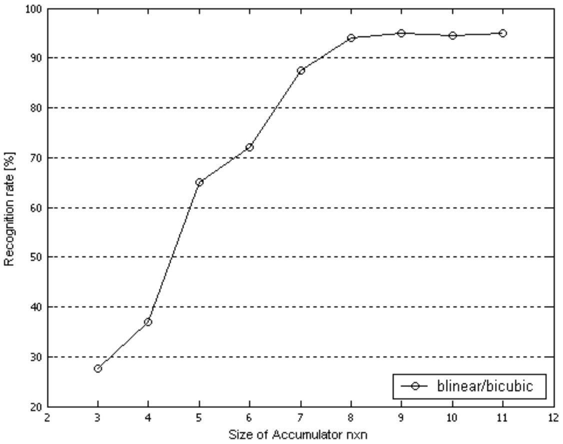


Fig 12. Influence of Accumulator size on recognition rate

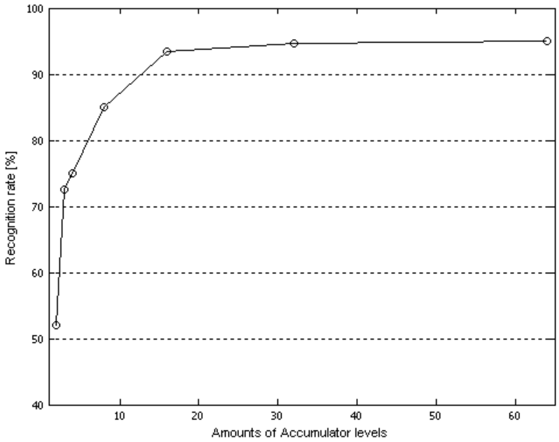


Fig 13. Influence of amount Accumulator levels on recognition rate

[8] D. Smith, “A Tutorial on Principal Components Analysis”, http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[9] R. D. Ledesma, “Determining the Number of Factors to Retain in EFA: an easy-to-use computer program for carrying out Parallel Analysis”, *Practical Assessment, Research & Evaluation*, Volume 12, Number 2, 2007.

[10] R. Keys, "Cubic convolution interpolation for digital image processing", *IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing*,1981.

[11] A. V. Oppenheim and R. W. Schaeffer, “Digital Signal Processing”, *Englewood Cliffs*, Prentice-Hall, 1975.

[12] S. Avidan and A. Shamir, “Seam Carving for Content-Aware Image Resizing”,*ACM Transactions on Graphics*, Volume 26, Number 3, 2007.

[13] *The UTF-8 encoding form, The UTF-8 encoding scheme.* “UCS Transformation Format 8,” defined in Annex D of ISO/IEC 10646:2003, technically equivalent to the definitions in the Unicode Standard, 2003.

[14] A. Aissaoui, “Normalised Fourier Coefficients for Cursive Arabic Script recognition”, Universite Mohamed, Morocco, 1999.

[15] C. Liu and H. Sako, “ Performance evaluation of pattern classifiers for handwritten character recognition”, *International Journal on Document Analysis and Recognition*, Springer-Verlag, 2002.

[16] G. Kaufmann and H. Bunke, “ Automated Reading of Cheque Amounts”, *Pattern Analysis & Applications*, Springer-Verlag 2000.

[17] A. Bellili and M. Giloux, ”An MLP-SVM combination architecture for handwritten digit recognition”, *International Journal on Document Analysis and Recognition*, Springer-Verlag, 2003.

[18] C. Høilund, “The Radon Transform”, Aalborg University, *VGIS*, 2007.

[19] O. Ramos and T. E. Valveny,"Radon Transform for Lineal Symbol Representation",*The Seventh International Conference on Document Analysis and Recognition*, 2003.

[20] S. Venturas and I. Flaounas, “Study of Radon Transformation and Application of its Inverse to NMR”, *Algorithms in Molecular Biology*, 2005.

[21] K. Kim “Face Recognition using Principle Component Analysis”, DCS, University of Maryland, College Park, USA 2003.

[22] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces”. *Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586--591, 1991.

[23] A. Asano, “Radon transformation and projection theorem”, Topic 5, *Lecture notes of subject Pattern information processing*, 2002 Autumn Semester, http://kuva.mis.hiroshima-u.ac.jp/~asano/Kougi/02a/PIP/

[24] A. Averbuch and R.R. Coifman, “Fast Slant Stack: A notion of Radon Transform for Data in a Cartesian Grid which is Rapidly Computible, Algebraically Exact, Geometrically Faithful and Invertible”, SIAM J. Scientific Computing, 2001

[25] E. Kupce and R. Freeman, “The Radon Transform: A New Scheme for Fast Multidimensional NMR”, *Concepts in Magnetic Resonance*, Wiley Periodicals, Vol. 22, pp. 4-11, 2004.

[26] A. K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall, 1989.

[27] Imaginis - Computed Tomography Imaging (CT Scan, CAT Scan), http://imaginis.com/ct-scan/how_ct.asp

ISBN 978-83-60810-14-9
ISSN 1896-7094

# The Quasi-One-Way Function and Its Applications to Image Watermarking

Kazuo Ohzeki
Shibaura Institute of Technology
Toyosu Koutou-ku , Japan
Email: ohzeki @ sic.shibaura-it.ac.jp

Engyoku Gi
Shibaura Institute of Technology
Toyosu Koutou-ku, Japan
Email: 10848@sic.shibaura-it.ac.jp

***Abstract*—This paper describes a one-way function for use in an image watermark to improve authentication ability. The one-way function is popular in cryptography and can prove encryption security. The existence of the one-way function has not been proved yet, while the public key encryption procedure is widely used in practical commerce. The authors proposed a quasi-one-way function, which is weakly defined for practical use. The differences between the strict one-way-function and the proposed quasi-one-way function are discussed. Applications of the quasi-one-way function to image watermarking are shown. Two watermarking systems with SVD method and Jordan canonical form are tried. The robustness of the proposed watermarking method is high.**

## I. Introduction

Watermarking is a prospective method to enable forensic tracking of content distribution [1]. A robust one-way operation is expected to be an important component of any system to improve security for authentication of watermarking. The so-called "inversion attack" on watermarked data is based on the addition property of embedding. For an image, embedding is usually done by add operation to cover data. If that addition can be subtracted inversely by an inverse element in that algebraic domain, the inversion attack can be committed easily on any newly determined watermark [2]. The problem originates from this algebraic system's inclusion of an inverse element. A one-way function is a breakwater to prevent inversion attacks and improve system security. Several one-way function candidates have been discussed in the encryption world. Though procedures based on the one-way function are widely used in practical commerce and authentication, the existence of the one-way function has not been proved by a mathematical method.

In this paper, we re consider such a quasi-one-way function and also compare it with the strict one-way function. Watermark embedding is an addition to a cover image. Transforming the cover image into some specific constrained region, the addition is restricted by the region rule. The restriction induces prohibition of subtraction of the data. In such an arrangement, a directional function with addition, whose subtraction can be difficult, is expected to be developed.

A survey paper pointed out the gap between theoretical and practical security. Information-theoretic models for security represent the worst-case, while practical applications exist for optimistic security [3]. Two major security methods are shown; spread-spectrum and asymmetric. This paper takes on a kind of asymmetric method. Four methods were pointed out for security establishment [4]. They are the use of a Trusted Third Party, the asymmetric watermarking scheme, watermark detection using a group of proxies and the Zero-knowledge watermarking detection protocol.

This paper presents a description of a new one-way property for image watermarking. The framework of the algebraic domain is an integer set of image data.

Singular Value Decomposition (SVD) diagonalizes matrix elements by multiplying orthogonal matrices to obtain zero values for off-diagonal elements. The addition of watermarking elements to off-diagonal positions is a quasi-one-way functional operation. The quasi-one-way operation means, in this discussion, that a forward operation result is easily obtained, but it is more difficult to find an inverse value from the result than a forward value. A strict one-way function is not known in mathematical formulation [5]. A quasi-one way operation is one method of realizing a very effective countermeasure against the so-called inversion attack. Many research papers have described inversion attacks, but the best method of dealing with them remains an open problem. Some methods embed a watermark into a random sequence instead of into normal images. One method theoretically embeds a watermark cryptographically. Then it uses a Zero-Knowledge detection method, which differs from an image watermark and is too vulnerable to small attacks.

Gorodetski et al. [5] introduced a watermarking method using SVD in 2001. Ganic [6] surveyed SVD-based watermarking methods in 2003. Many other papers have presented the use of SVD, but the problems of embedding methods remain. Features of SVD that have been discussed include robustness, combination with DCT, and wavelet transformation. A countermeasure to inversion attacks remains elusive. At present, we can not find an SVD watermarking method that features a quasi-one-way function to protect against inversion attacks.

In the study described in this paper, SVD is renovated from a mathematical perspective. Discussions of this matter are revised, and an improved SVD-based watermarking method is developed with a quasi-one-way function to cope

978-83-60810-14-9/08/$25.00 © 2008 IEEE

with inversion attacks. An outline of the SVD watermarking and the quasi-one-way function is proposed in [7]. In this paper, the difference between the strict one-way function and the proposed quasi-one-way function is discussed. Applications of the SVD to image watermarking with one-way directional property are also described. Improved results for a larger size of images compared to the previous experiments [7] are shown, together with experiments using the Jordan canonical form.

## II. Image Watermarking

### A. Inversion Attack

Embedding a watermark into a cover image usually involves an addition of the image and the watermark. As shown in Fig. 1, for an input image G and a watermark W, we obtain $G_w\ w=G+W$ . For this result, an attacker first makes a new his own new watermark W", which he derives by embedding a watermark information $W''$ into an image $G'$ to get $G_w\ '=G'+W''$ and subtracting image $G'$ to get $W''=G'w-G'$ . Using this watermark $W''$ , he declares that he had possessed another image $G_w -W''$ and then embedded watermark W" to get $(G_w -W'')+W''$ , which is the same data as $G_w$ . $G_w$ can have the watermark of $W''$ . Viewing this process, for an arbitrary watermarked image, another person can claim that he had embedded another watermark. This is an inversion attack. The inversion attack always works on the images embedded using a simple addition.

To prevent such inversion attacks it is necessary to introduce a new kind of one-way addition or to transform the embedding procedure into a region with a one-way operation.

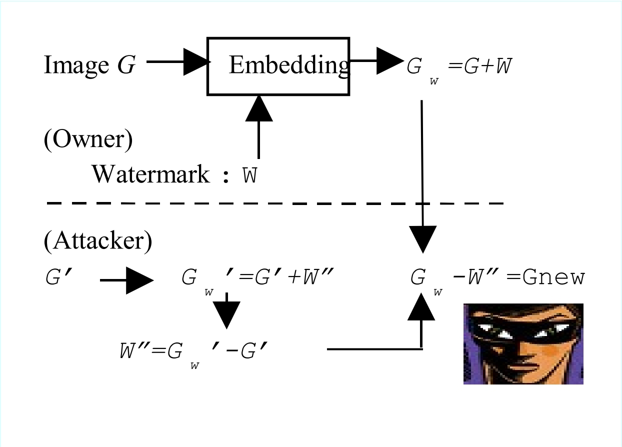


Fig 1. Inversion Attack.

### B. Authentication

For authentication of an image watermark, it is necessary to prove that a detected watermark is definitely the embedder's mark and that the procedure for detecting the watermark from the image is true. If we disclose the procedure to the public, it would mean that everyone knows the detection method and the embedding method and would be able to remove the watermark from that procedure. It is necessary to prove an ownership without disclosing the knowledge of how the watermark truly corresponds to the ownership, because it is also possible to remove the watermark from the image using such knowledge. A zero-knowledge watermark system was presented [5], however, the image is a random number.

A watermark suffers a high error rate as a transmission media, only a restricted number of bits are available for watermarking [8]. At present, it is difficult and impractical for the moment, to construct a watermarking authentication system with public key encryption and zero-knowledge proof. Hence, it would be practical to develop a watermarking system for an application so that watermarks can be embedded into images for web publication. The application would still require authentication at some prescribed level, and to attack such a system would entail certain costs, though it is not perfect to protect the ownership. Examples of authentication methods are to show matrices U and V of singular value decomposition, and to disclose the watermark to someone who can play a role of witness.

## III. One-Way Function For Watermarking

### A. Necessity For One-Way Function

Let us consider a one-way operation and the inversion attack in regard to the embedding process of a watermark. Given a watermark W and an embedded image $G_w$ $(G_w =G+W)$ , whose embedding function is $f$ , the inverse function of the embedding function $f$ is easily obtained as the inverse of the addition. In the case of embedding a watermark in a frequency domain, it is also easy to find the inverse function if the embedding method uses the standard Fourier transform. As long as the embedding process uses an addition for embedding, the inverse function can be easily found using a subtraction. Usually, image data are integer values between [0,255], and all watermarks are represented as additions. It is expected that a one-way operation for the embedding process will be introduced. A one-way function or determining a new algebraic field are candidates for this.

An example of realizing a kind of one-way function is presented [2]. An SVD watermarking system is shown in Fig. 2. An input Image $G$ is decomposed into a diagonal matrix $S$ by the singular value decomposition method using orthogonal matrices $U$ and $V$ . The matrix $S$ is diagonal, and its diagonal elements are singular values while the off-diagonal elements are zeroes. Then, let us consider a procedure in which adding a watermark $Ws$ , which has non-zero elements on off-diagonal positions, recovers image data from $S+Ws$ by multiplying $U$ and $V$ to get $G_w$ . If we re-decompose this embedded image $G_w$ again, we will obtain another set of $U'$ and $V'$ SVD matrices that are different from $U$ and $V$ . So, if we try to subtract $W_s$ from $G_w$ , we cannot get the original image $G$ . Furthermore, elements of the embedded image $G_w$ are usually truncated to integer values, which brings about a non-linear relationship to the watermark embedding system.

### B. Quasi-One-Way Function

The o ne-way function is formalized mathematically

A function $f:\{0,1\}^* \rightarrow \{0,1\}^*$ is called one-way if the following two conditions hold [9]:

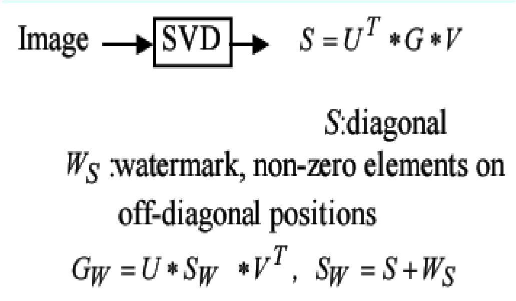


Image
SVD
$S = U^T * G * V$
S:diagonal
$W_S$ :watermark, non-zero elements on off-diagonal positions
$G_W = U * S_W * V^T$, $S_W = S + W_S$

Fig. 2 SVD Watermarking

1. Easy to Evaluate: There exists a probabilistic polynomial-time algorithm *A L* such that *A L* ( *x* ) = *f* ( *x* ) for every *x*.
2. Hard to Invert: For every probabilistic polynomial-time Turing machine *M* and for any polynomial *p(n)* , for all *n>N*,

$$\Pr\left(M\left[f\left(U_n\right),1^n\right]\in f^{-1}f\left(U_n\right)\cap\sum{}^n\right)<\frac{1}{p(n)} \qquad (1)$$

where, $U_n$ is a uniform probability distribution and the probability is taken over in {0,1}* and the coin tosses in *M* .

Several one-way function candidates have been discussed. It is computationally difficult to obtain an original number from a squared number or from a multiple of two prime numbers under a modulo rule, as indicated by the Robin function. Given a number, there is no fast algorithm of factorization of the number into prime factors. P ro ving the existence of a one-way function is still an open problem. However, difficulty in find ing prime factors is common in practical applications without mathematical verification. This indicates that there should be more sub-classes for practical applications below the polynomial time *(P)* . The term "quasi-one-way function" was mentioned by Whitfield Diffie in a paper [7] which said that "a quasi one-way function is not one-way in that an easily computed inverse exists. However, it is computationally infeasible , even for the designer, to find the easily computed inverse. Therefore a quasi one-way function can be used in place of a one-way function with essentially no loss in security ". Based on this concept, it is useful to newly create practical computational classes below polynomial. Here, we will extend the statement from computationally infeasible to having more complex mandatory operations than that for forward operation as an inverse function. It is a more realistic and constructive approach to weaken the definition and utilize the number of times of a finite computation.

Definition of a quasi-one-way function:

For a function *y=f(x)* , evaluating the minimum number of times of the forward operation *y=f(x)* and the inverse operation $x=f^{-1}(y)$ , a quasi-one-way function should have a larger number of inverse operations than forward operations, as shown by formula (2).

$$\min\left(Num\left(y=f\left(x\right)\right)\right)<\mathrm{Min}\left(Num\left(x=f^{-1}\left(y\right)\right)\right) \qquad (2)$$

It is recommended that the number of operations of the inverse function is noted.

### 1. C. SVD and Quasi-ONE-WAY FUNCTION

We will consider the relation between the SVD watermark embedding in Fig. 2 and the quasi-one-way function. If we are given a SVD watermark embedded image $G_w$ and its watermark $W_s$ , then to derive SVD decomposing matrices *U* and *V* may be possible by correcting SVD of $G_w$ using $W_s$ if we neglect the quanti z ing error of $G_w$ . However, the watermark embedded image $G_w$ is truncated to an integer value, which means $G_w$ loses some of the information required to recover the correct SVD decomposing matrices *U* and *V* . It is anticipated that the inverse function will require several times as many operations as the forward function. Concerning the image size of *nXn* , the number of forward operations is order $O(n^3)$ for a non-sparse matrix. The uniqueness of the SVD, which was already proved, governs the inverse operation to be difficult.

## IV. Singular Value Decomposition

### A. Method 1

In consideration of the preceding section, after embedding a watermark into the singular value matrix *S* , it is not necessary to re- apply SVD. Moreover , a quasi-one - way characteristic of SVD lies in the fact that the off-diagonal position values are zeros. N o positive or negative values exist in the off-diagonal positions . Therefore, to put a value on an off-diagonal position is a quasi-one-way operation. For that reason, adding watermark values at off-diagonal positions is an important quasi-one-way operation. As long as the quasi-one-way operation is effective, it is difficult to find another watermark with an appropriate pair of *U* and *V* of SVD for the embedded image, $G_W$ .

To improve the SVD-based watermarking method, it is merely necessary to remove the operation of the second SVD. The embedded watermark is *SS=S+aW* . Then, the inverse SVD is performed to obtain an embedded image,

$$G_W = U * SS * V^T$$

If another SVD is applied to this $G_W$ , then

$$G_W = U_W * S_W * V_W^T .$$

In general, $SS \neq S_W$ , $U \neq U_W$ , and $V \neq V_W$ .

It is noteworthy that *SS* has off-diagonal elements other than zero, although $S_W$ has only diago nal elements; from $S_W$, an embedded watermark cannot be detected.

On the other hand, the first and real owner who embedded the watermark can detect *SS* and *W* using *U, V* and *S* . This is "Method 1" of improved SVD-based watermarking with a quasi-one-way function.

However, this Method 1 present s a problem. Alt hough neither *SS* n or *W* can be derived directly from $G_W$ , another effective *U', V'* and *W* ', which m ight differ from the correc t *U, V* and *W* , can be computed using basic linear algebra. Consequently, an inversion attack can be made on this Method 1. Fo r example, after another SVD,

$$G_W = U_W * S_W * V_W^T .$$

Using a proper regular matrix *T*, $S_W = U_W^T * G_W * V_W$ can be modified, by multiplying *T* from the left and $T^{-1}$ from the right, to

$$T * U_W^T * G_W * V_W * T^{-1}$$

Putting $T_U = T, T_V = T^{-1}$, we obtain

$$T_U * S_W * T_V = T_U * U_W^T * G_W * V_W * T_V \quad , (3)$$

using a pair of example matrices for $T$, as

$$T_U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \varepsilon & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_U^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\varepsilon & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_V \quad .$$

Then, (3) is modified to yield the following [10].

$$T_U * S_W * T_V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \varepsilon & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & s_4 \end{bmatrix} * T_V$$

$$= \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & s_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \varepsilon s_1 - \varepsilon s_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= S_W^* \quad (4)$$

Subsequently, (4) is decomposed into a sum of two matrices as,

$$S_W^* = S_W^D + W' \quad ,$$

where $S_W^D$ is a diagonal component and $W'$ is an off-diagonal component.

On the other hand, (3) can be decomposed into a product of two matrices as follows:

Using $T_U * S_W * T_V = T_U * U_W^T * G_W * V_W * T_V$ , we obtain

$$(T_U * U_W^T)^{-1} * T_U * S_W * T_V * (V_W * T_V)^{-1} = I_W \quad .$$

Now we can find another SVD for the embedded image $G_W$ with re-defined decomposition matrices as

$$U^* = (T_U * U_W)^{-1} * T_U \quad , \quad V^{*T} = T_V * (V_W * T_V)^{-1} \quad .$$

The following (5) can be inferred from the above [10]. In fact, $T_U$ is a versatile matrix for any watermarked image embedded using Method 1.

$$U^* V^{*T} = (T_U * U_{W^T})^{-1} * T_U * T_V (V_W * T_V)^{-1} \quad (5)$$
$$= I.$$

### B. Method 2

An improved method was proposed to overcome the problem presented in the preceding section [10] . This Method 2 was devised because Method 1 includes the defect that using an appropriate orthogonal matrix $T$ , the diagonal matrix *S* can be transformed easily into another matrix with off-diagonal components , which breaks the quasi-one-way function. The diagonal matrix *S* and watermark matrix *W* are first reviewed to formulate Method 2.

For image matrix $G$ , SVD produces a pair of orthogonal matrices, $U$ and $V$ , where $U$ stands for column transformation and *V* is the row transformation. Multiplying an orthogonal matrix *"T"* to the diagonal singular matrix S can generate the watermark matrix *W*, which contains non-zero off-diagonal components. In fact, *SS* has two expressions,

$$SS = S + W$$

and

$$SS = T_U * S \quad .$$

Then, from $S+W= T_U * S$ ,

$W = (T_U - I) * S$ or $T_U = (S + W) * S^{-1}$

are obtained. Observing the latter formula, in the case of matrices S and $T_U$ in Method 1, *W* is not regular, and its diagonal components diminish. The rank of the watermark matrix *W* increased if the number of embedded watermarks increased . Increasing the rank of *W* tends to decrease the rank of $T_U$ .

Based on these considerations , Method 2 proposes a non-regular matrix $T_U$ . To reduce the rank of $T_U$ , a partial copy of S into *W* generates the linearly dependent matrix SS and consequently , $T_U = (S + W) * S^{-1}$ is also non-regular. In SVD, the rank of $U^* = T_U * U_W$ decreases and is not regular. By this operation, the embedded image $G_W$ is decomposed as,

$$G_W = U_W^{**} * S_W^{**} * V_W^{**} \quad .$$

The rank of $S_W^{**}$ , $U_W^{**}$ and $V_W^{**}$ decreases. Thereby it is computationally difficult to obtain regular matrices which match the original $G_W$ from these reduced rank matrices . The simplest example of *W* is,

$$W(i,i+1)=S(i,i) \ , \ W(i+1,i)=S(i+1,i+1).$$

A flowchart of Method 2 is shown in Fig. 3.

## V. Experimental Results

In this section, the proposed SVD-based watermarking algorithm is described using numerical data to confirm the operational methods in detail.

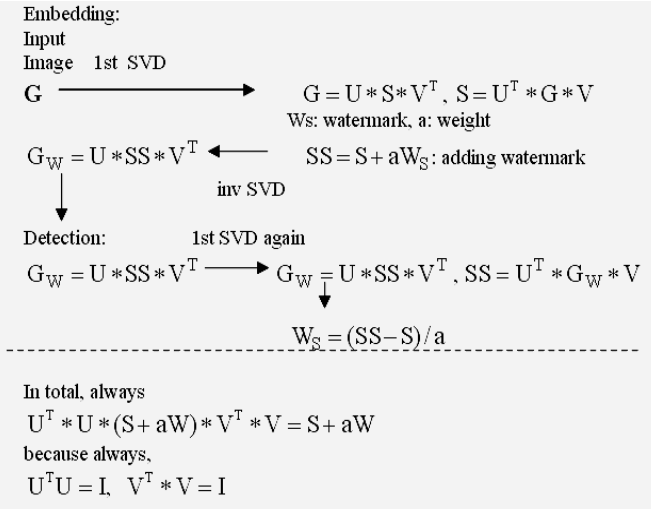


Fig 3.: Proposed SVD-Based watermarking embedding and detection.

### A. Basic Analysis

For a 4 × 4 image **G**, SVD is shown as the following.

$$\mathbf{G} = \begin{pmatrix} 132 & 122 & 114 & 108 \\ 122 & 116 & 110 & 106 \\ 110 & 116 & 106 & 107 \\ 104 & 107 & 109 & 99 \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} 447.9 & 0 & 0 & 0 \\ 0 & 13.0 & 0 & 0 \\ 0 & 0 & 6.5 & 0 \\ 0 & 0 & 0 & 1.1\ 5 \end{pmatrix}$$

$$\mathbf{U} = \begin{pmatrix} -0.533 & -0.652 & 0.108 & -0.528 \\ -0.507 & -0.252 & 0.003 & 0.824 \\ -0.490 & 0.415 & -0.747 & -0.172 \\ -0.468 & 0.582 & 0.656 & -0.112 \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} -0.524 & -0.827 & 0.114 & 0.168 \\ -0.515 & 0.118 & -0.439 & -0.727 \\ -0.490 & 0.407 & 0.769 & -0.051 \\ -0.469 & 0.370 & -0.450 & 0.664 \end{pmatrix}$$

Next, as a watermark matrix **W**, to make the second column of the singular value matrix conform to the third column,

$$\mathbf{W}(2,3)=\mathbf{S}(2,2),\ \mathbf{W}(3,2)=\mathbf{S}(3,3).$$

are processed. Then **SS**=**S**+**W** is obtainable.

$$\mathbf{W}= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 6.5 & 0 \\ 0 & 13.0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{SS} = \begin{pmatrix} 447.9 & 0 & 0 & 0 \\ 0 & 13.0 & 6.5 & 0 \\ 0 & 13.0 & 6.5 & 0 \\ 0 & 0 & 0 & 1.15 \end{pmatrix}$$

Next, the embedded image $\mathbf{G}_w = \mathbf{U} * \mathbf{SS} * \mathbf{V}^{\mathrm{T}}$ is obtained by inverse SVD using **SS**, **U**, **V**.

$$\mathbf{G}_w= \begin{pmatrix} 130.3 & 124.0 & 111.3 & 110.4 \\ 121.8 & 116.7 & 108.7 & 106.8 \\ 118.3 & 113.7 & 104.1 & 102.2 \\ 97.4 & 106.3 & 115.4 & 100.4 \end{pmatrix}$$

An attacker will attempt to find the singular value matrix **SS** to obtain the embedded watermark matrix **W**. To do so, $\mathbf{G}_w$ is decomposed as,

$$\mathbf{G}_w = \mathbf{Uw}^{**} * \mathbf{Sw}^{**} * \mathbf{Vw}^{**\,\mathrm{T}},$$

$$\mathbf{Uw}^{**} = \begin{pmatrix} -0.532 & -0.387 & -0.454 & -0.601 \\ -0.509 & -0.175 & 0.840 & -0.071 \\ -0.490 & -0.232 & -0.278 & 0.793 \\ -0.466 & 0.875 & -0.107 & -0.070 \end{pmatrix}$$

$$\mathbf{Vw}^{**}= \begin{pmatrix} -0.524 & -0.684 & 0.197 & -0.468 \\ -0.516 & -0.099 & -0.747 & 0.407 \\ -0.490 & 0.711 & -0.030 & -0.503 \\ -0.469 & 0.130 & 0.634 & 0.601 \end{pmatrix}$$

$$\mathbf{Sw}^{**}= \begin{pmatrix} 447.4 & 0 & 0 & 0 \\ 0 & 20.5 & 0 & 0 \\ 0 & 0 & 1.40 & 0 \\ 0 & 0 & 0 & 0.19 \end{pmatrix}$$

The rank of $\mathbf{Sw}^{**}$ is theoretically 3, although it seems to be 4, the fourth value is extremely small, almost zero. Most parts of singular values are sufficiently large, indicating that Method 2 is valid for use in pictures of a general nature.

Many other images were tested with application of SVD. Table 1 shows the maximum and minimum of the singular values of matrix **S**. The maximum values are from **S**(1,1) and the minimum values are from the last non-zero diagonal component. The images in Table 1 are all natural ones; the rank of **S** is the same value as the image size, except for the circles image. It is an artificial image, not a natural one. Such images usually contain the same value in lines as background parts, and the rank might decrease. For all other images shown in Table 1, SVD operations were all well performed: the ranks of the singular matrix *S* are all the same value as the image size, implying that the proposed method is stable in decomposition for many natural scene images.

### B. Jordan Canonical Form

Another basic analysis with the Jordan canonical form was carried out. For an image matrix *G* as,

$$G= \begin{bmatrix} 236 & 10 & 11 & 12 \\ 10 & 177 & 12 & 10 \\ 10 & 12 & 174 & 12 \\ 10 & 12 & 14 & 222 \end{bmatrix}$$

Eigenvalues are well obtained because this matrix is regular

$$\Lambda = P^{-1} * G * P = \begin{bmatrix} 249.3 & 0 & 0 & 0 \\ 0 & 217.3 & 0 & 0 \\ 0 & 0 & 179.2 & 0 \\ 0 & 0 & 0 & 163.3 \end{bmatrix}$$

Table 1
Ranks of Images and the Maximum and Minimum Values of Singular Values . (girl * is a partial image from the center of an image girl from (126,126) to (129,129). Circles** is an artificial computer graphic image.)

| image name | size | rank of S | singular value | |
|---|---|---|---|---|
| | | | maximum | minimum |
| girl_* | 4×4 | 4 | 447 | 0.2 |
| car | 240×240 | 240 | 29761 | 0.4 |
| girl | 256 × 256 | 256 | 16511 | 0.3 |
| couple | | 256 | 10003 | 0.1 |
| lena | | 256 | 31956 | 0.0 1 |
| peppers | | 256 | 27203 | 0.1 |
| circles ** | | 125 | 28509 | 40 |
| lena | | 512 | 38638 | 0.0 03 |

where $P=\begin{bmatrix} 0.805 & 0.603 & 0.163 & -0.023 \\ 0.218 & -0.076 & -0.731 & -0.624 \\ 0.222 & -0.100 & -0.562 & 0.779 \\ 0.505 & -0.788 & 0.351 & -0.054 \end{bmatrix}$

To modify the matrix to have the Jordan canonical form, let the third and fourth components of eigenvalues be the same for having double root and putting a value “1” on the off-diagonal right-hand position of the third eigenvalue. This operation forces the matrix to have the Jordan canonical form that contains a watermark. Let the Jordan’s mark $W$ be

$$W=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

then, the embedded matrix is,

$$\Lambda_m w=\begin{bmatrix} 249.3 & 0 & 0 & 0 \\ 0 & 217.3 & 0 & 0 \\ 0 & 0 & 179.2 & 1 \\ 0 & 0 & 0 & 179.2 \end{bmatrix}=\Lambda_m+W \quad .$$

Inverse transformation derives the embedded image matrix as,

$$GwI=R(P*\Lambda_m w*P^{-1})=\begin{bmatrix} 236 & 10 & 11 & 12 \\ 10 & 184 & 4 & 11 \\ 10 & 5 & 183 & 11 \\ 10 & 12 & 14 & 222 \end{bmatrix}$$

where $R$ (*) represents rounding.

Because the uniqueness of the Jordan canonical form is guaranteed by mathematical theory, the obtained image data matrix is the Jordan matrix.

Multiplying a pair of matrices $P$ and $P^{-1}$ can detect the Jordan’s mark.

$$P^{-1}*G_{w^I}*P=\begin{bmatrix} 249.3 & 0 & -0.1 & 0.3 \\ 0 & 217.3 & 0 & -0.2 \\ -0.2 & 0.1 & 179.6 & 1.41 \\ -0.2 & 0.6 & 0 & 178.8 \end{bmatrix} \quad .$$

To get the Jordan canonical form from the truncated matrix, $G_w{}^I$ could not be done by the change of rounding of integer data. Actually, the result of getting the Jordan canonical form from $G_w{}^I$ is,

$$\Lambda_m wI=\begin{bmatrix} 249.3 & 0 & 0 & 0 \\ 0 & 217.3 & 0 & 0 \\ 0 & 0 & 179.4 & 1 \\ 0 & 0 & 0 & 179.0 \end{bmatrix}$$

which has no double root any more and is not a Jordan matrix, but rather a normal regular matrix .

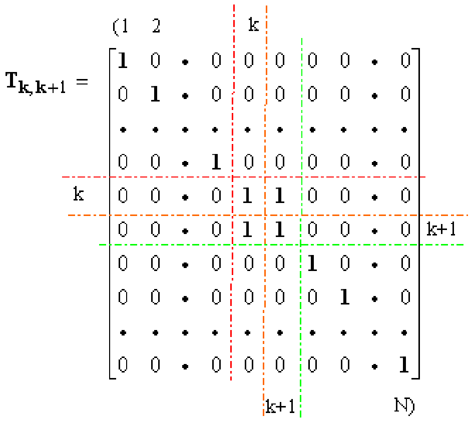
This Jordan canonical form method is theoretically interesting. However, error sensitivity is large. Therefore, it might not be robust. The example above shows a difference of the embedded mark from 1.00 to 1.41. Embedding trials were done for several image datasets, but the example’s performance and robustness remained unsatisfactory. Eigenvalues can be complex even for real integer data.

### C. Embedding Watermarks

Based on the above considerations, watermark-embedding experiments were carried out using Method 2. Fig. 4 shows that the embedding was realized by multiplying a matrix $T_{k,k+1}$ . The embedded images were modified using JPEG compression. Detection ratios are shown in Fig. 5 . Table 2 shows their embedded position and singular values. The actual added data are,

*SS(k,k+1)=S(k;1,k+1)* and *SS(k+1,k)=S(k,k)*.

The detection rule for these experiments is that all singular values are maintained at specified levels. The specified levels are half of the original values. The detection ratios shown in Fig. 5 are normalized by the original values. The dotted line at 50% represents the borderline between detectable and non-detectable. Without JPEG compression, 100% detection is achieved because only a small fractional error exists. Compression ratios of 11-13 are the maximum for detection. In the figure, *SVD_min* means the minimum value of two modified SVD values by JPEG at $S$ ( $k,k$ ) and $S$ ( $k+1,k+1$ ). These values are maintained as larger than the borderline for JPEG compression ratio of 30. In addition, *WM_min* means the minimum value of the embedded two watermarks modified by JPEG at $S$ ( $k,k+1$ ) and $S(k+1,k$ ). These values are kept larger than the borderline for JPEG compression ra-



$$\mathbf{T_{k,k+1}} = \begin{array}{c|cccccccccc|c} & (1 & 2 & & & k & & & & & & \\ \hline & 1 & 0 & \cdot & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & \\ & 0 & 1 & \cdot & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & 0 & 0 & \cdot & 1 & 0 & 0 & 0 & 0 & \cdot & 0 & \\ k & 0 & 0 & \cdot & 0 & 1 & 1 & 0 & 0 & \cdot & 0 & \\ & 0 & 0 & \cdot & 0 & 1 & 1 & 0 & 0 & \cdot & 0 & k+1 \\ & 0 & 0 & \cdot & 0 & 0 & 0 & 1 & 0 & \cdot & 0 & \\ & 0 & 0 & \cdot & 0 & 0 & 0 & 0 & 1 & \cdot & 0 & \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & 0 & 0 & \cdot & 0 & 0 & 0 & 0 & 0 & \cdot & 1 & \\ \hline & & & & & & k+1 & & & & N) & \end{array}$$

Fig 4. Embedding Matrix $T_{k,k+1}$ .

tio 11 or 13. *Ripple _ Max* means the maximum value among all other elements in the neighbouring area except the four elements; *S(k,k)* , *S(k,k+1)* , *S(k+1,k)* and *S(k+1,k+1)* . The *Ripple_Max* area values are originally zeroes and are modified by JPEG compression. The *Ripple_Max* is generally small. For JPEG compression ratio 30, the *Ripple_Max* is less than 15%.

Table:2
Embedded positions and Singular Values.

| Image | Embedded Position(1) k | SVD Value | Embedded Position(2) k+1 | SVD Value |
|---|---|---|---|---|
| girl | 50 | 205.06 | 5 1 | 195.68 |
| couple | 5 0 | 197.62 | 51 | 189.70 |
| lena | 85 | 202.40 | 86 | 201.46 |
| peppers | 74 | 204.88 | 75 | 199.37 |
| lena2 | 80 | 406.16 | 81 | 395.78 |

### *D. Detection M ethod 2*

The detection method above is an elementary version. An improved detection method (DM 2) can be devised from Fig. 5. Room exists between ripples and the 50% detection boundary line. Ripples around diagonal positions with SVD values on them are generally small. Therefore, the detection threshold level *Lw* of embedded watermarks is expected to be smaller than that of a logical value. Furthermore, the SVD value threshold level *Ls* ha s been adjusted. Fig. 6 depicts the results for other images. Detection r atios are improved by this change of threshold. The image quality of JPEG-coded images can be recognized easily for compression ratio s higher than 15 for these experiments. Some degradation is apparent for the compression ratio of 10. The coded images will be shown at oral presentation. Fig. 7 shows a comparison of robustness related to the image size. Images of 256 × 256 were used for the first experiment. An image that wa s twice as large as that used in the previous experiment ha d larger singular values and show ed higher robustness for embedding. Detection ratios of lena_506 we re much higher than the images of 256. For a JPEG compression ratio of 30, the detection wa s well carried out: the larger the image, the higher the obtained robustness.

## VI. Conclusions

Quasi-one-way functions are proposed for watermarking applications. Singular Value Decomposition and Jordan canonical form are introduced for obtaining computationally asymmetrical structures. Jordan canonical form is said to be difficult to calculate precisely for a large matrix. Error sensitivity is large. For real values image data eigenvalues can be complex numbers. Imaginary parts of a pair of conjugate complex eigenvalues resulting from double rooting processing in making Jordan canonical form are likely to be large. Reorganizing the SVD-based watermarking system, an improved SVD-based watermarking method was developed with a quasi-one-way function by reduced rank matrix. The developed matrix with reduced-rank for watermark embedding in singular value decomposition cannot be restored using simple methods, such as simply multiplying a matrix. So-called inversion attacks cannot be activated as long as the proposed quasi-one-way operation framework holds. They would require computationally complex procedures to derive the exact set of singular value decomposition matrices. Results of experiments underscore the effectiveness of the proposed system for smaller sizes of image data. For larger sizes of image data, the detection ratio increases under JPEG compression attacks. In the future, an iterative operation for increasing computational complexity must be investigated. In addition, detection performance can be improved using a smart er and more complicated algorithm

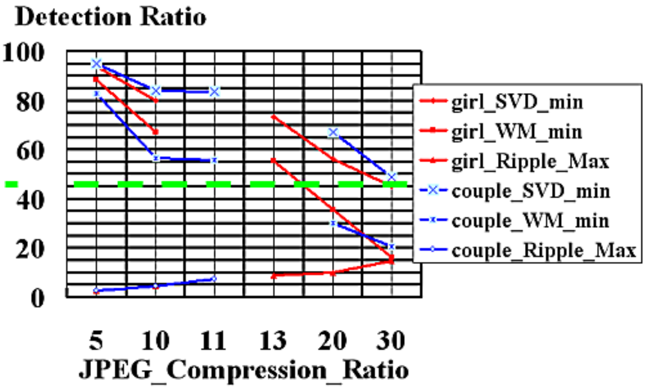


Fig 5. Detection Ratios Depending on Compression

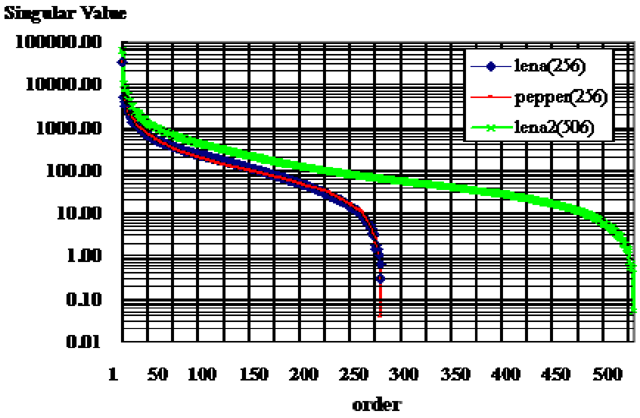


Fig 6. Singular values for larger size of image .

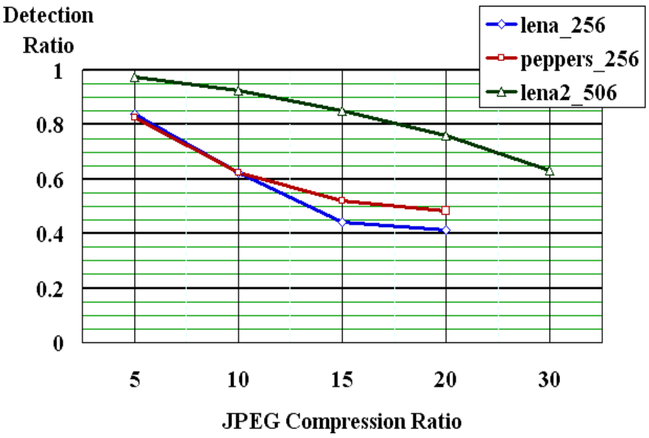


Fig 7 Comparison of detection ratios between large and small sizes of images.

## References

[1] Martin Schmucker ed., “First Summary Report on Forensic Tracking”, IST-2002-507932 ECRYPT, D.WVL.7-1.1.pdf , Jan. 2005.

[2] Scott Craver et al., “Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications”, IEEE J. SAC Vol.16,No.4 pp. 573-586, May 1998.

[3] Luis Pérez-Freire et al., “Watermarking security: a survey”. *Transactions on Data Hiding and Multimedia Security I*, 4300:41-72, October 2006.

[4] Qiming Li and EeChien Chang, "ZeroKnowledge Watermark Detection Resistant to Ambiguity Attacks", Proc. ACM Multimedia and Security Workshop pp. 158-163, Sept. 2006.

[5] Goldreich, Oded, “Foundations of Cryptography: A Primer”, Now Publishers Inc 2005.

[6] Ganic, E.et al., “An Optimal Watermarking Scheme Based on Singular Value Decomposition”. Proc. of CNIS. 85-90. 2003.

[7] W. Diffie et al., "New Directions in Cryptography" IEEE Trans.-IT Vol.22, 6, pp.644-654, 1976.

[8] Deepa Kundur, "Authentication Watermarking," ECRYPT *(CMS-2005)*, Sept. 22, 2005.

[9] S. Aida et al., “Average-Time Analysis of One-Way Functions”, IEICE Tech. Rept. COMP99-28, pp.47-54, 1999.

[10] Kazuo Ohzeki and Masaru Sakurai, “SVD-Based Watermark with Quasi-One-Way Operation by Reducing a Singular Value Matrix Rank”, Proc. of e-forensics, Tech B4. WM, 1. Jan., 2008.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Question generation for learning evaluation

Liana Stanescu, Cosmin Stoica Spahiu,
Anca Ion, Andrei Spahiu
University of Craiova, Faculty of
Automation, Computers and Electronics
Email: {Stanescu_Liana, Stoica_Cosmin,
Anca.Ion}@software.ucv.ro

***Abstract*—In the last decade the electronic learning became a very useful tool in the students' education from different activity domains. The accomplished studies indicated that the students substantially appreciate the e-learning method, due to the facilities: the facile information access, a better storage of the didactic material, the curricula harmonization between universities, personalized instruction. The paper presents a software tool that can be used in the e-learning process in order to automatically generate questions from course materials, based on a series of tags defined by the professor. The Test Creator tool permits generation of questions based on electronic materials that students have. The solution implies teachers to have a series of tags and templates that they manage. These tags are used to generate questions automatically.**

## I. Introduction

IN THE last decade, the electronic learning became a very useful tool in the students' education from different activity domains. The accomplished studies indicate that the students substantially appreciate the e-learning method, due to the facilities offered [1], [2], [3], [6], [7]: the facile information access, a better storage of the didactic material, the curricula harmonization between universities, personalized instruction, informational content standardization, real time access to qualitative information resources and friendly interfaces. They don't consider it as a replacement of the traditional learning that has other advantages [8].

As it is known, an essential aspect in the learning process (either electronic or traditional) is the possibilities to evaluate the students. It is very important both for professor and student to test the understanding degree of the course. One of the best possibilities is to ask questions from the studied course. It is tested this way the degree of understanding of each studied material and the integration of new knowledge with the previous ones (that should already be known). These facts will have as a result an in-depth understanding of the learning materials. The studies showed that a high number of questions from the same subject in-depth understanding. That is why a series of questions can be found as exercises at the end of many high rated courses.

The paper presents the implementation of a tool that can generate questions automatically based on the tags defined by the professor. He can add new tags, delete the existing ones and generate questions specifying the part of the course that should be used for questions.

The structure of the paper is: in the second part it is presented the related work. In the third part it is presented the application, the architecture of the application, and in the last part, the conclusions.

## II. Related Work

Taking into consideration the high number of learning material existing in electronic format, the importance of the testing and evaluation systems has increased. Most of these systems use tests that were generated by teachers that permit a good evaluation and pursuance of the student evolution.

In the last years, new preoccupations appear for automatic question generation. It's a subclass of Natural Language Generation (NLG) that is very important in a series of areas as: learning environment, data mining or information extraction. For example in [1] it is introduced a template-based approach to generate questions on four types of entities. It is considered that his approach failed in producing questions that can enhance the students' knowledge level. The authors present in [2] an interesting solution to the problem of presenting students with dynamically generated browser-based exams with significant engineering mathematics content. They introduce WTML (Web Testing Markup Language), which is an extension of HTML. A very interesting approach is found in [3]. Here, the main idea is to generate the questions automatically based on question templates which are created by training on many medical articles. This idea has advantages (easiness in building medical learning system, no additional work to build the question database or grading), but also disadvantages: the generated questions are factual and maybe less meaningful than the manual questions, time consuming to parse the articles and obtain the semantic interpretation, missing some important information.

Taking into account the advantages and disadvantages of the presented solutions, we tried to design and implement a software instrument (Test Creator) that permits generation of questions based on electronic materials that students have. The solution implies teachers to have a series of tags and templates that they have to manage. These tags can be used to generate questions automatically.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## III. Question Generation

### A. General Description

In this paragraph there are presented concepts and the working style for the Test Creator software tool. The main window is organized in several sub-windows, each of them permitting some operations in a very simple manner. It is easily compatible with any learning domain: engineering, medical, or economical. In order to generate questions, based on a specific course material, there should be followed 3 steps:

1. Defining tags or questions categories
2. Defining templates for a specific tag
3. Parsing the text in order to generate the templates.

The basic idea for generating questions based on the course material is to define a list of tags, chosen by the teacher. Each tag represents a class of questions with similar formatting that are applicable for certain theoretical notions from the course.

For example, the <DEFINE> tag will be used to formulate some questions where the student has to define some concepts. The <EXAMPLE> tag will be used for questions where it should be presented an example for a specific concept.

In the TestCreator software tool there is a sub-window where there are presented the tags that already exists in the database. They can be updated using insert/delete operations. The most important thing is that the teacher has total freedom in choosing these tags, as he considers being most suitable for his course domain.

For each tag, the teacher defines one or several forms of a question, suitable for a specific category. These forms of the questions were called templates (figure 1).

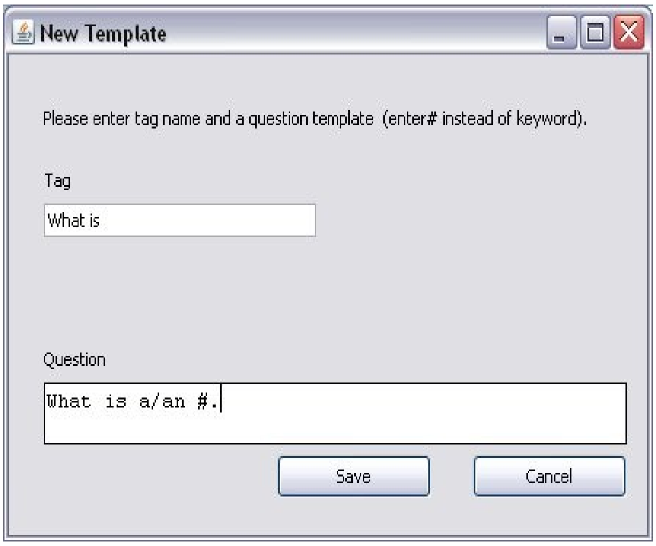
For example, for the <DEFINE> tag it can be defined the following template: "DEFINE #". For the tag <WHAT IS>, the template can be "What is a/an #". The use of "#" sign represents for this application the reserved word or phrase to which it is applied the tag.

One of the problems was to have several forms for the same tag in order to give the possibility to the teacher to create questions that are correct from the syntactic and semantic point of view. Another situation that appears frequently refers to the possibility to formulate the same questions both in the native language and in an international language.

The final step is represented by the questions generation. The professor has to load first the course material in the main window. Then, in order to generate questions, he has to select the keyword or key-phrase from the text and then to select the tag and template. The "#" sign will be replaced with the chosen word/phrase from the course.

For example, for the DEFINE # template, and the keywords "Boyce-Codd normal form" there will be generated the question: DEFINE Boyce-Codd normal form.

As soon as it was generated, the question is displayed in another window of the software tool. The teacher has to decide if it is correct and if it should be kept and stored in the database (figure 2).



Please enter tag name and a question template (enter# instead of keyword).

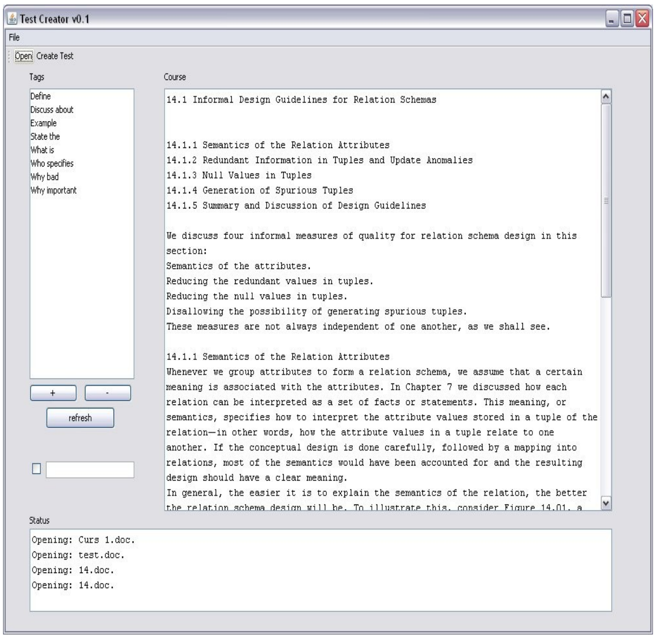
Tag

What is

Question

What is a/an #.

Fig. 1 Defining the question template



Tags

Define
Discuss about
Example
State the
What is
Who specifies
Why bad
Why important

Course

```
14.1 Informal Design Guidelines for Relation Schemas



14.1.1 Semantics of the Relation Attributes
14.1.2 Redundant Information in Tuples and Update Anomalies
14.1.3 Null Values in Tuples
14.1.4 Generation of Spurious Tuples
14.1.5 Summary and Discussion of Design Guidelines

We discuss four informal measures of quality for relation schema design in this
section:
Semantics of the attributes.
Reducing the redundant values in tuples.
Reducing the null values in tuples.
Disallowing the possibility of generating spurious tuples.
These measures are not always independent of one another, as we shall see.

14.1.1 Semantics of the Relation Attributes
Whenever we group attributes to form a relation schema, we assume that a certain
meaning is associated with the attributes. In Chapter 7 we discussed how each
relation can be interpreted as a set of facts or statements. This meaning, or
semantics, specifies how to interpret the attribute values stored in a tuple of the
relation—in other words, how the attribute values in a tuple relate to one
another. If the conceptual design is done carefully, followed by a mapping into
relations, most of the semantics would have been accounted for and the resulting
design should have a clear meaning.
In general, the easier it is to explain the semantics of the relation, the better
the relation schema design will be. To illustrate this, consider Figure 14.01, a
```

Status

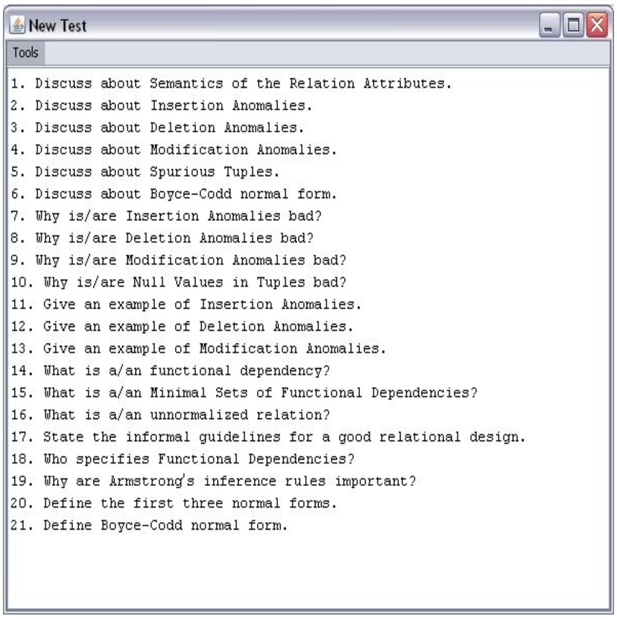
```
Opening: Curs 1.doc.
Opening: test.doc.
Opening: 14.doc.
Opening: 14.doc.
```

Fig. 2 The main window of the software tool

Another window of the application is the one in figure 3, where can be displayed all the questions generated by the teacher for the course material currently loaded in the system. These questions are managed by the teacher giving him the possibility to delete, update and store them in the database. He will also have the possibility to reload a course material and see the questions generated for it.

### B. The Structure of the Database

The entity-relation model of the database used by the software tool is presented in the figure 4. The Tags table is the one that stores information about existing query categories. The Templates table has a connection of 1:m with the table Tags.



1. Discuss about Semantics of the Relation Attributes.
2. Discuss about Insertion Anomalies.
3. Discuss about Deletion Anomalies.
4. Discuss about Modification Anomalies.
5. Discuss about Spurious Tuples.
6. Discuss about Boyce-Codd normal form.
7. Why is/are Insertion Anomalies bad?
8. Why is/are Deletion Anomalies bad?
9. Why is/are Modification Anomalies bad?
10. Why is/are Null Values in Tuples bad?
11. Give an example of Insertion Anomalies.
12. Give an example of Deletion Anomalies.
13. Give an example of Modification Anomalies.
14. What is a/an functional dependency?
15. What is a/an Minimal Sets of Functional Dependencies?
16. What is a/an unnormalized relation?
17. State the informal guidelines for a good relational design.
18. Who specifies Functional Dependencies?
19. Why are Armstrong's inference rules important?
20. Define the first three normal forms.
21. Define Boyce-Codd normal form.

Fig. 3 The questions generated by the system

It is used to store for each query category, different forms of the queries. The tables Course and Chapters stores information about electronic courses and their chapters. Actually these chapters will be loaded into the application in order to generate questions from them. Between tables Chapters and templates there is a m:m connection. In the relational model this will lead to a new relation, called Questions where it is stored the effective content of the generated query

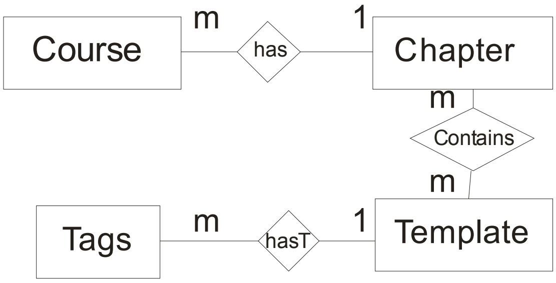


Fig. 4 The Entity-Relationship model for database structure

## IV. Software Architecture

The document type that is the most used is the MS Word .doc type. For this we have implemented a class (POIDoc) that opens and returns the content of a MS Word document. This class uses Apache POI libraries: poi-3.0.1.jar, poi-contrib-3.0.1.jar, poi-scratchpad-3.0.1.jar.

The POI project consists of APIs for manipulating various file formats based upon Microsoft's OLE 2 Compound Document format using pure Java. We can read and write MS Word, Excel files using Java. POI is your Java Word solution as well as your Java Excel solution. HWPF is the name of our port of the Microsoft Word 97 file format to pure Java. It does not support the new Word 2007, .docx file format, which is not OLE2 based.

For basic text extraction, we make use of org.apache. poi.hwpf.extractor.WordExtractor. It accepts an input stream or a HWPFDocument. The getText() method can be used to get the text from all the paragraphs, or getParagraphText() can be used to fetch the text from each paragraph in turn. The other option is getTextFromPieces(), which is very fast, but tends to return things that aren't text from the page.

## V. Concept Maps Applied for Questions Generation

### A. General Description

Concept maps are a result of Novak and Gowin's (1984) research into human learning and knowledge construction. Novak (1977) proposed that the primary elements of knowledge are concepts and relationships between concepts are propositions. Novak (1998) defined concepts as "perceived regularities in events or objects, or records of events or objects, designated by a label." Propositions consist of two or more concept labels connected by a linking relationship that forms a semantic unit.

Concept maps are a graphical two-dimensional display of concepts (usually represented within boxes or circles), connected by directed arcs encoding brief relationships (linking phrases) between pairs of concepts forming propositions. The simplest concept map consists of two nodes connected by an arc representing a simple sentence such as 'flower is red,' but they can also become quite intricate.

One of the powerful uses of concept maps is not only as a learning tool but also as an evaluation tool, thus encouraging students to use meaningful-mode learning patterns.

Concept mapping may be used as a tool for understanding, collaborating, validating, and integrating curriculum content that is designed to develop specific competencies. Concept mapping, a tool originally developed to facilitate student learning by organizing key and supporting concepts into visual frameworks, can also facilitate communication among faculty and administrators about curricular structures, complex cognitive frameworks, and competency-based learning outcomes. To validate the relationships among the competencies articulated by specialized accrediting agencies, certification boards, and professional associations, faculty may find the concept mapping tool beneficial in illustratingrelationships among, approaches to, and compliance with competencies [9].

According to this approach, the responsibility for failure at school was to be attributed exclusively to the innate (and, therefore, unalterable) intellectual capacities of the pupil. The learning/ teaching process was, then, looked upon in a simplistic, linear way: the teacher transmits (and is the repository of) knowledge, while the learner is required to comply with the teacher and store the ideas being imparted [10].

It should be made a very important distinction between rote learning and meaningful learning.

Meaningful learning requires three conditions:

1. The material to be learned must be conceptually clear and presented with language and examples relatable to the learner's prior knowledge. Concept maps can be helpful to meet this condition, both by

identifying large general concepts held by the leaner prior to instruction of more specific concepts, and by assisting in the sequencing of learning tasks though progressively more explicit knowledge that can be anchored into developing conceptual frameworks;

2. The learner must possess relevant prior knowledge. This condition can be met after age 3 for virtually any domain of subject matter, but it is necessary to be careful and explicit in building concept frameworks if one hopes to present detailed specific knowledge in any field in subsequent lessons. We see, therefore, that conditions (1) and (2) are interrelated and both are important;
3. The learner must choose to learn meaningfully. The one condition over which the teacher or mentor has only indirect control is the motivation of students to choose to learn by attempting to incorporate new meanings into their prior knowledge, rather than simply memorizing concept definitions or propositional statements or computational procedures. The indirect control over this choice is primarily in instructional strategies used and the evaluation strategies used. Instructional strategies that emphasize relating new knowledge to the learner's existing knowledge foster meaningful learning. Evaluation strategies that encourage learners to relate ideas they possess with new ideas also encourage meaningful learning. Typical objective tests seldom require more than rote learning [7].

### *B. Concept maps for queries generation*

The concept map helps the professor to have a better overview of the course and what aspect he should pay attentions. If he generate first a concepts map he will be able to define tags for every concept and implicit to have questions for each concept.

Before defining the tags position in the text the professor has to specify a list of concepts and the connections between them. The list of concepts that will be included is chosen by the professor. He will decide what the most important concepts in the course are and which are not.

There are many tools that generate concept maps. The most used are:

- C-TOOLS – Luckie (PI), Implemented to the University of Michigan NSF grant. It is available for download to the address:

http://ctools.msu.edu/ctools/index.html

- TPL-KATS – Implemented to the University of Central Florida (e.g., Hoeft, Jentsch, Harper, Evans, Bowers, & Salas, 1990). TPL-KATS: concept map, a computerized knowledge assessment tool. Computers in Human Behavior, 19 (6), 653-657.
- SEMNET – Downloadable from: http://www.semanticresearch.com/about/

The steps that should be followed are:

1. The specialist will use one of these specialized tools to generate a concept map for the course material, similar to the one presented in figure 5.
2. Load the course in the system
3. Add/Delete existing tags templates to be according to his needs.
4. For each concept in the map define one or several templates that will be applied.
5. Generate questions. For each edge in the graph it will correspond a certain number of quiz questions.

The algorithm transforming the Concept Map into General Graph is strait forward. Each proposition becomes an edge with a weight assigned by domain knowledge expert. In this way it was obtained the Binary Search Tree General Graph. Once the General Graph has been set, up the professor has to set up the quiz questions for the chapter. For each edge in the graph it will correspond a certain number of quiz questions.

Once the quiz questions have been set up, for each student there may be constructed the learner's associated graph.

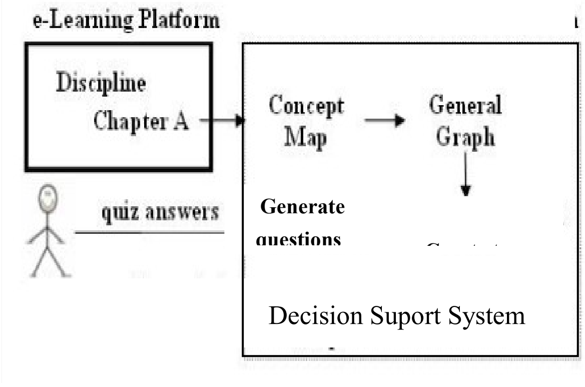


Fig. 6 Functionality of Test Creator tool using on Concept Maps.

This graph will have associated with the edges the history of correct and wrong answered questions. The Calculus engine will reconstruct an Annotated Concept Map which will present to the learner the current status of his knowledge level at Concept level. In this way, the learner will have an exact overview of his knowledge level regarding that chapter. The Annotated Concept Map may represent the important information for learner in having a decision regarding which part of the chapter needs more study.

## VI.Conclusions and Future Work

The paper presented a software tool that can be used in the e-learning process in order to generate automatically questions from course materials. The professor can define tags and templates that will be used in query generation.

The Test Creator tool permits generation of questions based on electronic materials that students have. The solution implies teachers to have a series of tags and templates that they have to manage. These tags can be used to generate questions automatically.

In order to find the most important part in a course material and it will be created a list of concepts and a Concept map. For these steps of the process it should be used a third party software tool specialized in concept maps aspects.

It is preferable to include concept maps concepts in the learning process for two reasons:

- you can be sure that you will have questions about all important concepts existing in the course
- you can monitor learning activity to be sure that students learn meaningful and not only several separate aspects, with no connection between.

## References

[1] A. Andrenucci, .Sneiders, E., "Automated Question Answering: Review of the Main Approaches", in *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, July 4-7, Sydney, Australia, IEEE, Vol. 1, 2005, pp.514-519.

[2] J. McGough, J. Mortensen, J. Johnson, S. Fadali "A web-based testing system with dynamic question generation". *LNCS 1611-3349*, 2008, pp. 242-251.

[3] W. Wang, H. Tianyong, L. Wenyin, "Automatic Question Generation for Learning Evaluation in Medicin", in *LNCS Volume 4823*, 2008, pp. 242-251.

[4] L. Vecchia, M. Pedroni, "Concept Maps as a Learning Assessment Tool" in *Issues in Informing Science and Information Technology*, Volume 4.

[5] E. McDaniel, B. Roth, M. Miller, "Concept Mapping as a Tool for Curriculum Design", in *Issues in Informing Science and Information Technology*.

[6] C. Jonathan, A. Gwen, E. Maxine Eskenazi, "Automatic question generation for vocabulary assessment", in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing,* p.819-826, October 06-08, 2005, Vancouver, British Columbia, Canada.

[7] D.D. Burdescu, M. C. Mihaescu, "Building a decision support system for students by using concept maps", in *Proceedings of International Conference on Enterprise Information Systems (ICEIS'08)*, Barcelona, Spain, 2008.

[8] E. McDaniel, B. Roth, M. Miller, "Concept Mapping as a Tool for Curriculum Design", in *Issues in Informing Science and Information Technology.*

# International Conference on Principles of Information Technology and Applications

INTERNATIONAL Conference on Principles of Information Technology and Applications (PITA '08) is the most general Event of the IMCSIT. PITA invites contributions presenting research results in any area of computer science or information technology, which are not covered by other Events of the IMCSIT.

Topics include but are not limited to:

- Computer systems development and maintenance
- Foundations of computer science
- System dependability
- Modeling and simulation
- Formal methods
- Software engineering
- Parallel and distributed processing
- IT applications
- E-country, E-learning
- IT Management

We are especially interested in any novel, may be controversial solutions. The interdisciplinary contributions and papers that hardly classify to typical topics of conferences are invited as well.

We would like to extend a particularly warm welcome to young researchers whose research is supported by EU, national and local grants. We are interested in papers concerning theory and applications as well as case studies of successful transfer of technology from academia to industry.

## Organizing Committee

**Anna Derezinska,** Warsaw University of Technology, Poland

**Tomasz Pelech-Pilichowski,** AGH University of Science and Technology, Poland

## Program Committee

**Witold Abramowicz,** The Poznan University of Economics, Poland

**Rimantas Butleris,** Kaunas University of Technology, Lithuania

**Witold Byrski,** AGH University of Science and Technology, Poland

**Wojciech Cellary,** The Poznań University of Economics, Poland

**Zbigniew Czech,** Silesian University of Technology, Poland

**Marek Druzdzel,** University of Pittsburgh, USA

**Jozef Goetz,** University of La Verne, USA

**Anna Hać,** Telcodia Technologies, Inc., USA

**Michal Iglewski,** Université du Québec en Outaouais, Canada

**Ryszard Janicki,** McMaster University, Canada

**Stanisław Jarząbek,** National University of Singapore, Singapore

**Mieczysław Kłopotek,** Institute of Computer Science, Polish Academy of Sciences, Poland

**Mieczyslaw Kokar,** Northeastern University, USA

**Beata Konikowska,** Institute of Computer Science, Polish Academy of Sciences, Poland

**Michael L. Korwin-Pawlowski,** Université du Québec en Outaouais, Canada

**Maciej Koutny,** University of Newcastle upon Tyne, UK

**Andrzej Lingas,** Lund University, Sweden

**Boleslaw Mikolajczak,** University of Massachusetts, USA

**Tomasz Müldner,** Acadia University, Canada

**Mieczyslaw Muraszkiewicz,** Warsaw University of Technology, Poland

**Thomas F. Piatkowski,** Western Michigan University, USA

**Andrzej Sluzek,** Nanyang Technological University, Singapore

**Michael Sobolewski,** Texas Tech University, USA

**Janusz Sosnowski,** Warsaw University of Technology, Poland

**Helena Szczerbicka,** Leibniz Universität Hannover, Germany

**Bogdan Wiszniewski,** Gdansk University of Technology, Poland

**Wlodek M. Zuberek,** Memorial University, Canada

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Correctness issues of UML Class and State Machine Models in the C# Code Generation and Execution Framework

Anna Derezińska
Institute of Computer Science Warsaw University of Technology, ul. Nowowiejska 15/19 00-665
Warszawa, Poland
Email: A.Derezinska@ii.pw.edu.pl

Romuald Pilitowski
Institute of Computer Science Warsaw University of Technology, ul. Nowowiejska 15/19 00-665
Warszawa, Poland

***Abstract*** **— Model driven approach for program development can assist in quick generation of complex and highly reliable applications. Framework for eXecutable UML (FXU) transforms UML models into C# source code and supports execution of the application reflecting the behavioral model. The framework consists of two parts code generator and run time library. The generated and executed code corresponds to structural model specified in class diagrams and behavioral model described by state machines of these classes. All single concepts of state machines included in the UML 2.0 specification (and further) are taken into account, including all kinds of events, states, pseudostates, submachines etc. The paper discusses the correctness issues of classes and state machine models that have to be decided in the framework in order to run a model-related and high quality C# application. The solution was tested on set of UML models.**

## I. Introduction

MODEL Driven Engineering (MDE) represents software development approaches in which creation and manipulation of models should result in building of an executable system [1].

Industrial product development puts a lot of attention on fast implementation of the needed functionalities. Model-driven approach to program development offers a promising solution to these problems. The complex behavioral models can be designed and verified at the early stages of the whole product creation cycle and automatically transformed into the code preserving the desired behavior.

State machines, also in the form of statecharts incorporated in the UML notation [2], are a widely used concept for specification of concurrent reactive systems. Proposal for execution of behavioral UML models suffers from the problem that no generally accepted formal semantics of UML models is available. Therefore, validation of UML transformation and model behavior depicted in the resulting code is difficult. Rather than completely formalizing UML models, we try to deal with selected aspects of the models.

Checking of models is important in Model Driven Architecture (MDA) approaches [3], [4] where new diagrams and code are automatically synthesized from the initial UML model: all the constructed artifacts would inherit the initial inconsistency [5].

Inconsistency and incompleteness allowed by UML can be a source of problems in software development. A basic type of design faults is concerned with the well-formedness of diagrams [2]. Typically, completeness of a design requires that introduced model elements are specified with their features and usage of one element can imply a usage of another, directly related model element. In the current modeling CASE tools some completeness conditions can be assured automatically (e.g., default names of roles in associations, attributes, operations etc.). Incompleteness of models can be to be strongly related to their inconsistency, because it is often impossible to conclude whether diagrams are inconsistent or incomplete [6]. Therefore, within this paper we will refer to model defects as to correctness issues.

The Framework for eXecutable UML (FXU) offers a foundation for applying MDA ideas in automation of software design and verification. The FXU framework was the first solution that supported generation and execution of all elements of state machine UML 2.0 using C# language [7]. In order to build an application reflecting the modeled classes and their behaviors specified by state machines, we resolved necessary semantic variation points [8]. Semantic variation points are aspects that were intentionally not determined in the specification [2] and its interpretation is left for a user.

It was also necessary to provide some correctness checking of a model. This paper is devoted to these issues. To present potential problems we selected one target application environment, i.e., creation of application in C# language. The verification of an input UML model is based on a set of hard coded rules. Some of the rules are general and can be applied for any object-oriented language, as they originate directly from the UML specification [2]. Other rules are more environmental specific because they take also into account the features of the target language - C#. The verification is performed during transformation of class and state machine models into the corresponding code; it is so-called static verification. Other set of rules is used during execution of the code corresponding to given state machines; so-called dynamic verification. For all correctness rules the appropriate reaction on the detected flaws were specified.

This work was supported by the Dean of the Department of Electronics and Information Technology, Warsaw University of Technology under grant no 503/G/1032/4300/008

In the next section we discuss the related works. Next, the FXU framework, especially solutions used for state machines realization, will be presented. In Sec. IV we introduce correctness issues identified in the transformation process and during execution of state machines. Remarks about experiments performed and the conclusions finish the paper.

## II. Related Work

A huge amount of research efforts is devoted to formalization of UML models, specification of their semantics and verification methods [9]-[13]. However they are usually not resolving the practical problems which are faced while building an executable code, because of many variation semantic points of the UML specification.

An attempt for incorporation of different variation points into one solution is presented in [14]. The authors intend to build models that specify different variants and combine them with the statechart metamodel. Different policies should be implemented for these variants.

Our work relates also to the field of consistency of UML models. The consistency problems in UML designs were extensively studied in many papers. It could be mentioned workshops co-located to the Models (former UML) series of conferences, and other works [5], [6], [15]-[17].

An interesting investigation about defects in industrial projects can be found in [18]. However the study takes into account only class diagrams, sequence diagrams and use case diagrams, mostly the relations among elements from different diagram types. The state machines were not considered.

Solutions to consistency problems in class diagrams were presented in [19]. The problem refers to constrains specifying generalization sets in class diagram, which is still not commonly used in most of UML designs.

Current UML case tools allow constructing incorrect models. They provide partial checking of selected model features, but it is not sufficient if we would like to create automatically a reliable application. More comprehensive checking can be found in the tools aimed at model analysis. For example, the OO design measurement tool SDMetrics [20] gives the rules according to which the models are checked. We used the experiences of the tool (Sec. IV), but it does not deal with state machine execution nor with C# language.

Many modeling tools have a facility of transforming the models into code in different programming languages. However, the most of them consider only class models. We compared functionality of twelve tools that could also generate code from state machines. Only few of them took into account more complex features of state machines, like choice pseudostates, deep and shallow history pseudostates, deferred events or internal transitions. The most complete support for state machines UML 2.0 is implemented in the Rhapsody tool [21] of IBM Telelogic (formerly I-Logix). However it does not consider C# language.

Different approaches to generation of the code from behavioral UML models can be used. The semantics of a state machine can be directly implemented in the generated code [22]. Another solution is usage of a kind of a run-time environment, for example a run-time library as applied in the FXU framework.

The consistency problems remain also using tools for building executable UML models [23], [24]. Different subsets of UML being used and we cannot assure that two interchanged models will behave in the same way. Specification of a common subset of UML specialized for execution is still an open idea.

## III. Code Generation and execution in FXU

Transformation of UML models into executable application can be realized in the following steps.

1. A model, created using a CASE modeling tool, is exported and saved as an XML Metadata Interchange (XMI) file.
2. The model (or its parts) is transformed by a generator that creates a corresponding code in the target programming language.
3. The generated code is modified (if necessary), compiled and linked against a Runtime Library. The Runtime Library contains realization of different UML meta-model elements, especially referring to behavioral UML models.
4. The final application, reflecting the model behavior, can be executed.

It should be noted, that steps 1) and 2) can be merged, if the considered code generator is associated with the modelling tool.

The process presented above is realized in the FXU framework [7]. The target implementation language is C#. The part of UML model taken into account comprises classes and state machines. The input models are accepted in UML2 format, an XMI variant supported by Eclipse. Therefore it is not directly associated with any modelling tool. However, all experiments mentioned in Sec. V were performed with UML models created using IBM Rational Software Architect [25].

The FXU framework consists of two components - FXU Generator and FXU Runtime Library. The Generator is responsible for realization of step 2. The FXU Runtime Library includes over forty classes that correspond to different elements of UML state machines. It implements the general rules of state machine behavior, independent of a considered model, e.g., processing of events, execution of transitions, entering and exiting states, realization of different pseudostates. It is also responsible for the runtime verification of certain features of an executed model.

Transforming class models into C# code, all model elements are implemented by appropriate C# elements. The template of a resulting programming class can be found in [7]. Principles of code generation from the class models are similar to other object-oriented languages and analogues to solutions used in other tools.

A distinctive feature of FXU is dealing with all UML state machine elements and their realization in C# application. Therefore we present selected concepts of state machines with their implementation in C#. We point out different C# specific mechanisms used in the generated application. Using selected solutions we would like to obtain an efficient and reliable application.

State machines can be used at different levels of abstraction. They can model behavior of an interface, a component, an operation. Protocol state machines are intended to model protocols. The primary application of behavioral state machine in an object-oriented model is description of a class. A class can have attributes keeping information about a current state of an object. Classes have operations that can trigger transitions, send and receive events. Therefore, we assumed that the code will be generated and further executed only for behavioral state machines that are defined for certain classes that are present in the structural model.

An exemplary UML model is shown in Fig. 1. A given class has an attribute, four operations and its behavior specified by a state machine. The state machine consists of simple state S1 and complex state S2 including two orthogonal regions. In guard conditions and triggers the operations and attribute of the class are used. Extracts of the C# code corresponding to the example and created by the FXU generator are given in the Appendix.

For any state machine of a class, a new attribute of *StateMachine* type is created. Each class having a state machine has also two additional methods *InitFXU* and *StartFXU*. Method *InitFXU* is responsible for creation and initialization of all objects corresponding to all elements of state machine(s) associated with the class, such as regions, states, pseudostates, transitions, activities, events, triggers, guards, actions, etc. Method *StartFXU* is used for launching a behavior of state machine(s).

Any state can have up to three types of internal activities *do, entry, exit.* The activities of a state are realized using a delegate mechanism of C#. Three methods *DoBody, EntryBody* and *ExitBody* with empty bodies are created for any state by default. If an activity exists a corresponding method with its body is created, using information taken from the model. Applying delegate mechanism allows defining the methods for states without using of inheritance or overloaded methods. Therefore the generated code can be simple, and generation of a class for any single state can be avoided. A state machine is not generated as a state design pattern [26], because we would like to prevent an explosion of number of classes.

Three transition kinds can be specified for a transition, *external, internal* and *local* transitions. Triggering an internal transition implies no change of a state, exit and entry activities are not invoked. If an external transition is triggered it will exit its source state (a composite one), i.e. its exit activity will be executed. A local transition is a transition within a composite state. No exit for the composite (source) state will be invoked, but the appropriate exits and entries of the substates included in the state will be executed.

A kind of a transition can be specified in a model, but in praxis this information is rarely updated and often inaccurate. Therefore we assumed that in case of composite states a kind of generated transition is determined using a following heuristics:

- If the target state is different than the source state of a transition and the source state is a composite state, the transition is external.
- Else, the transition is defined in a model as internal it is treated as an internal transition.
- Otherwise, the transition is local.

A transition can have its guard condition and actions. They are created similarly to activities in states, using delegate mechanism of C#. If a body of an appropriate guard condition or action is nonempty in a model, it is put in the generated code. It should be noted that verification of logical conditions written in C# is postponed to the compilation time.

States, pseudostates, transitions and events are created as local variables. Signals are treated in different way. They are created as classes, because they can be generalized and specialized building a signals hierarchy. If a certain signal can trigger an event also all signals that are its descendants in the signal hierarchy can trigger the same event. This feature of signals was implemented using the reflection mechanism of C# [27].

Events should have some identifiers in order to be managed. Change events and call events are identified by unique natural numbers assigned to the events. A time event is identified by a transition which can be triggered by this event. A completion event is identified by a state in which the event was generated. Finally, for a signal event the class of the signal, i.e., its type, is used as its identifier.

There are some elements of a UML model that include a description in a form not precisely specified in the standard, but dependent on a selected notation, usually a programming

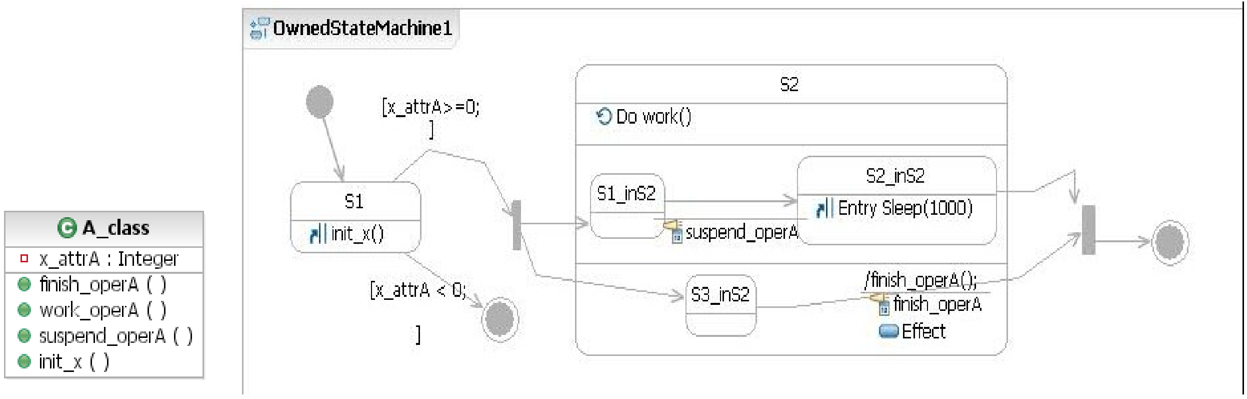


Fig. 1 Example - a class and its state machine

language. There are, for example, guard conditions, implementation of actions in transitions or in states, body of operations in classes. They can be written directly in a target implementation language (e.g., C#). During code generation these fragments are inserted into the final code. Verification of the syntax and semantics of such code extracts is performed during the code compilation and execution according to a selected programming language.

Interpreting different concepts of state machines we can use parallel execution. In the FXU RunTime Library it is implemented by multithreading. Multithreading is used for processing of many state machines which are active in the same time, e.g., state machines of different classes. It is used also for handling submachine states and orthogonal regions working within states, and for other processing of events. In the Appendix, parts of an output trace generated during execution of the exemplary state machine (Fig. 1) are shown. We can observe different threads, identified by number in brackets, that were created to deal with encountering events. For example, realization of transition from the pseudostate fork to substate S3_in S2 launched thread "[11]". Thread "[12]" was created to implement transition from the fork pseudostate to substate S1_inS2. In other execution run of the application the numbers and ordering of threads can be different.

Event processing during state machine execution is performed according to the rules given in UML specification [2]. Basic algorithms of FXU realization, like execution of a state machine, entry to a state, exit from a state, were presented in [7]. For every state a queue was implemented that pools incoming events. Events can be broadcasted or sent directly to the selected state machines. Events trigger transitions that have an active source state and their guard conditions evaluate to true. If many transitions can be fired, transition priorities are used for their selection. We had proposed and implemented an extended definition of transitions priority, in order to resolve all conflicts in case many transitions can be fired. This could not be achieved based only on the priority definition given in [2]. The detailed algorithm of selecting non-conflicting transitions can be found in [8]. Also resolving of other variation points, especially dealing with entering and exiting orthogonal states, is shown in [8].

## IV. Verification of model correctness

While generating valid C# code from UML class and state machine diagrams the certain conditions should be satisfied. There are many possible shortcomings present in the models that are not excluded by the modeling tools, or should be not prohibited due to possible model incompleteness at different evolution stages. They were analyzed taking into account the practical weaknesses of model developers.

The prepared correctness rules were based on three main sources: the specification of UML [2], the rules discussed in related works and other comparable tools, in particular in [20], and finally the own study, especially taking into account the features of C# language - the target of the model transformation [27].

Various shortcomings can be detected during different steps of application realization (Sec. 3). Many of them can be identified directly in the model, and therefore detected during model to code transformation step (step 2). Verification of such problems will be called static, as it corresponds to an automated inspection of a model. Other flaws are detected only during execution of the resulting application (step 4). Such dynamic verification will be completed by the appropriate classes of the FXU Runtime Library.

In tables I-III defects identified in classes and state machines are presented. The last column shows severity associated to the shortcomings. Three classes of severity are distinguished. If a defect detected in a model is called as *critical* the model is treated as invalid and the code generation is interrupted without producing the output. Later cases are clas-

Table I.
Defects detected in Uml class diagrams (static)

| No | Detected defects | Reaction | Severity |
|---|---|---|---|
| 1 | A generalization of an interface from a class was detected | Stop code generation | critical |
| 2 | A name of an element to be generated (e.g. a class, an interface, an operation, an attribute) is a keyword of C# language | Stop code generation | critical |
| 3 | A class relates via generalization to more than one general class | Stop code generation | critical |
| 4 | A cycle in class generalization was detected | Stop code generation | critical |
| 5 | A name of an element to be generated is missing | Generate the element pattern without its name. The element name has to be supplemented in the generated code. | medium |
| 6 | A name of an element to be generated is not a valid C# name. It is assumed that white characters are so common shortcoming that they should be automatically substituted by an underline character. | As above | medium |
| 7 | An interface visibility is *private* or *protected* . | Use *package* visibility . | low |
| 8 | A class visibility is *private* or *protected* . | Use *package* visibility . | low |
| 9 | An interface is *abstract.* | Treat the interface as no abstract. | low |
| 10 | An interface has some attributes. | Ignore attributes of the interface. | low |
| 11 | An interface has nested classes | Ignore classes nested in the interface. | low |
| 12 | A class that is no *abstract* has abstract operations. | Treat the class as *abstract*. | low |

TABLE II.
DEFECTS DETECTED IN UML STATE MACHINES (STATIC)

| No | Detected defects | Reaction | Severity |
|---|---|---|---|
| 1 | A cycle in signal generalization was detected | Stop code generation | critical |
| 2 | A signal inherits after an element that is not another signal | Stop code generation | critical |
| 3 | A signal relates via generalization to more than one general signal | Stop code generation | critical |
| 4 | A region has more than one initial pseudostate | Stop code generation | critical |
| 5 | A state has more than one deep history pseudostate or shallow history pseudostate | Stop code generation | critical |
| 6 | There are transitions from pseudostates to the same pseudostates (different than a choice pseudostate) | Stop code generation | critical |
| 7 | There are improper transitions between orthogonal regions | Stop code generation | critical |
| 8 | A transition trigger refers to an nonexistent signal | Stop code generation | critical |
| 9 | An entry point, join or initial pseudostate has no incoming transition or more than one incoming transition | Stop code generation | critical |
| 10 | A deep or shallow history pseudostate has more than one outgoing transition | Stop code generation | critical |
| 11 | A transition from an entry/exit point to an entry/exit point | Stop code generation | critical |
| 12 | An exit point has no any incoming transition | Stop code generation | critical |
| 13 | Transitions outgoing a fork pseudostate do not target states in different regions of an orthogonal states | Stop code generation | critical |
| 14 | Transitions incoming to a join pseudostate do not originate in different regions of an orthogonal state | Stop code generation | critical |
| 15 | There is a transition originating in an initial pseudostate or a deep/shallow history pseudostate and outgoing a nested orthogonal state | Stop code generation | critical |
| 16 | The region at the topmost level (region of a state machine) has no initial pseudostate | Warn a user | medium |
| 17 | A transition outgoing a pseudostate has a trigger | Ignore the trigger | medium |
| 18 | A tgransition outgoing a pseudostate (different from a choice or junction vertex) has a nonempty guard condition | Ignore the guard condition | medium |
| 19 | A transition targeting a join pseudostate has a trigger or nonempty guard condition | Ignore the trigger and/or condition | medium |
| 20 | A trigger refers to a non-existing operation | The transition will be generated but it cannot be triggered by this event | medium |
| 21 | A trigger refer to an abstract operation or to an operation of an interface | as above | medium |
| 22 | A time event is deferred | Treat the event as not being deferred | medium |
| 23 | A final state has an outgoing transition | Warn a user | medium |
| 24 | A terminate pseudostate has an outgoing transition | Warn a user | low |

sified as *medium* and *low*. In both cases the code generation is proceeded, although for *medium* severity it can require corrections before compilation. In all cases information about all detected shortcomings is delivered to a user. A detailed reaction to the found defect is described in the third column. While assigning severity levels and reactions to given defects we took into account general model correctness features but also requirements specific for C# applications.

### *A. Verification of Class Models*

Class diagrams describe a static structure of a system, therefore many their features can be verified statically before code generation. Table I summaries defects that are checked during static analysis of UML class models. It was assumed that some improvements can be added more conveniently in the generated code than in a model. The class models can be incomplete to some extend and we can still generate the code. Admission of certain model incompleteness can be practically justifiable because of model evolution.

It should be noted that not all requirements of generated code are checked by the generator. Some elements are verified later by the compiler. It concerns especially elements that are not directly defined by the UML specification, like the bodies of operations.

### *B. Verification of State Machines*

Similarly to class diagrams, different defects of state machines can be detected statically in the models. They are listed in Tab. II. Static detection of shortcomings in state machines is realized twice. First, it is made before model to source transformation (step 2). Second correctness checking is fulfilled before state machine execution. It is a part of step 4, during the initialization of the structure of a state machine.

For example, a static verification can be illustrated using a state machine from Fig. 1. Transition outgoing state *S3_inS2* has an event trigger - calling of an operation *finish_operA()*. However, this transition targets the join pseudostate. Therefore neither a trigger nor a guard condition can be associated with the transition. It violates the correctness rule 18 (Tab. II). This model flaw is quite often and is not critical. The

TABLE III.
DEFECTS DETECTED IN UML STATE MACHINES (DYNAMIC)

| No | Detected defects | Reaction | Severity |
|---|---|---|---|
| 1 | There is no enabled and no "else" transition outgoing a choice or junction pseudostate | Suspend execution - terminate | critical |
| 2 | A deep or shallow history pseudostate was entered that has no outgoing transitions and is "empty", i.e. either a final state was a last active substate or the state was not visited before | Suspend execution - terminate | critical |
| 3 | More than one transition outgoing a choice or junction pseudostate is enabled | Select one enabled transition and ignore the others | medium |
| 4 | There is no enabled transition outgoing a choice or junction pseudostate and there is one or more "else" transition outgoing this pseudostate | Select onr "else" transition and ignore other transitions | medium |
| 5 | More than one transition outgoing the same state is enabled | Select one transition and ignore the others | medium |

trigger will be omitted in the generated code and the designer will be warned about this exclusion.

State machines model system behavior; therefore not all their elements can be statically verified. A part of defects is detected dynamically, i.e., during execution of state machines. For example, a situation that two enabled transitions are outgoing the same choice pseudostate can be detected after evaluation of appropriate guard conditions, namely during program execution. Defects detected dynamically in state machines are presented in Tab. III.

## V. EXPERIMENTS

The presented approach for building the C# code and executing the automatically created applications was tested on over fifty models. The first group of ten models was aimed at classes. In experiments the correct and incorrect constructions encountering in class diagrams were checked, concerning especially association and generalization. Moreover, two bigger projects were tested. The first one was a part of MDA project called Acceleo [28]. The model described a design of a web page. The second one presented a metamodel of an object-oriented modeling language [29].

Models from the next group (above forty models) comprised different diagrams, including both classes and state machines. All possible constructs of UML 2.x state machines were used in different situations in the models. The biggest design included five state machines with about 80 states and 110 transitions, using complex and orthogonal states, different kinds of pseudostates and submachine states.

The programs realizing state machines were run taking into account different sequences of triggering events. The behavior modeled by state machines was observed and verified using detailed traces generated during program runs. They helped to test whether the obtained program behavior conforms to desired state machine semantics. For complex models, filtered traces that included selected information were also used.

The performed experiments have showed that an application realizing a behavior specified in state machine models can be developed in an effective and reliable way.

## CONCLUSION

In this paper we discussed the problems of creation of valid C# applications realizing ideas modeled by classes and their state machines. Different C# mechanisms were effectively used for implementation of the full state machine model defined in the UML 2.x specification. We showed which correctness issues of models have to be checked during model transformation (static verification) and during application execution (dynamic verification). The detailed correctness rules help a developer to cope with possible flaws present in UML models. In the difference to other tools, using FXU the state machines including any complex features can be effectively transformed into corresponding C# application. The tool support speeds up building of reliable applications including complex behavioral specifications. It can be especially useful for developing programs in which nontrivial state machines are intensely used, e.g., dependable systems, embedded reactive systems.

## APPENDIX

The appendix includes extracts of C# code generated for an exemplary class and its state machine shown in Fig. 1. Code of class operations are omitted (line 3). Method *InitFxu()* creates appropriate structure of the state machine. Method *StartFxu()* initializes behavior of the state machine.

```
1 public class A_class   {
2   private int x_attrA;
3   // operations of A_class (omitted)
4   StateMachine sm1 = new
     StateMachine("OwnedStateMachine1");
5 public void InitFxu(){
6   Region r1 = new Region("Region1");
7   sm1.AddRegion(r1);
8   InitialPseudostate v2 = new
              InitialPseudostate("");
9   r1.AddVertex(v2);
10  FinalState v3 = new FinalState("");
11  r1.AddVertex(v3);
12  State v4 = new State("S1");
13  v4.EntryBody = delegate(){ init_x();  };
14  r1.AddVertex(v4);
15  State v5 = new State("S2");
16  v5.DoBody = delegate(){ work_operA(); };
17  r1.AddVertex(v5);
18  Region r2 = new Region("Region1");
19  v5.AddRegion(r2);
20  Region r3 = new Region("Region2");
21  v5.AddRegion(r3);
22  State v6 = new State("S2_inS2");
23  v6.EntryBody = delegate()
    {System.Threading.Thread.Sleep(10000); };
24  r2.AddVertex(v6);
```

```
  State v7 = new State("S1_inS2");
  r2.AddVertex(v7);
  State v8 = new State("S3_inS2");
  r3.AddVertex(v8);
  Fork v9 = new Fork("");
  r1.AddVertex(v9);
  FinalState v10 = new FinalState("");
  r1.AddVertex(v10);
  Join v11 = new Join("");
  r1.AddVertex(v11);
  Transition t1 = new Transition(v2, v4);
  Transition t2 = new Transition(v4, v9);
  t2.GuardBody = delegate(){return x_attrA>=0;};
  Transition t3 = new Transition(v4, v10);
  t3.GuardBody = delegate(){return x_attrA<0;};
  Transition t4 = new Transition(v6, v11);
  Transition t5 = new Transition(v7, v6);
  t5.AddTrigger(new CallEvent("suspend_operA",
                                         1));
  Transition t6 = new Transition(v8, v11);
t6.AddTrigger(new CallEvent("finish_operA",
                                           2));
  t6.ActionBody = delegate(){finish_operA(); };
  Transition t7 = new Transition(v9, v8);
  Transition t8 = new Transition(v9, v7);
  Transition t9 = new Transition(v11,v3);
} //End of InitFXU
public void StartFxu()
  {   sm1.Enter(); }
}
```

Fragments of a detailed execution trace of the exemplary state machine (Fig. 1) are shown below. Time stamps of all log items are omitted for the brevity reasons. The trace was created under condition of two call events occurrences, suspend_operA() and finish_operA(). A number in brackets denotes a number of a thread that realizes a considered part of machine execution.

```
[1] WARN  - State diagram <OwnedStateMachine1>: Entered.
[1] INFO  - State diagram <OwnedStateMachine1>: Execution of entry-activity started. State is now active.
[1] DEBUG - State diagram <OwnedStateMachine1>: Execution of entry-activity finished.
[7] INFO  - Initial pseudostate <OwnedStateMachine1::Region1{::UnNamedVertex}>: Entered.
[7] DEBUG - Transition from Initial pseudostate <OwnedStateMachine1::Region1{::UnNamedVertex}> to State <OwnedStateMachine1::Region1::S1>: Traversing started.
[7] INFO  - State <OwnedStateMachine1::Region1::S1>: Execution of entry-activity started. State is now active.
        (...)  //part omitted
[3] DEBUG - State diagram <OwnedStateMachine1>: Completion event <> generated by State <OwnedStateMachine1::Region1::S1> has been dispatched.
[9] DEBUG - State <OwnedStateMachine1::Region1::S1>: Execution of exit-activity started.
[9] INFO  - State <OwnedStateMachine1::Region1::S1>: Execution of exit-activity finished. State is now inactive.
[10] DEBUG - Transition from State <OwnedStateMachine1 ::Region1::S1> to Fork <OwnedStateMachine1::Region1 {::UnNamedVertex}>: Traversing started.
[10] INFO  - Fork <OwnedStateMachine1::Region1 {::UnNamedVertex}>: Entered.
[11] DEBUG - Transition from Fork <OwnedStateMachine1 ::Region1{::UnNamedVertex}> to State <OwnedStateMachine1:: Region1::S2::Region2:: S3_inS2>: Traversing started.
[11] INFO  - State <OwnedStateMachine1::Region1::S2>: Execution of entry-activity started. State is now active.
[11] DEBUG - State <OwnedStateMachine1::Region1 ::S2>: Execution of entry-activity finished.
[13] INFO  - State <OwnedStateMachine1::Region1::S2>: Execution of do-activity started.
[13] DEBUG - State <OwnedStateMachine1::Region1 ::S2>: Execution of do-activity finished.
[11] INFO  - State <OwnedStateMachine1::Region1::S2:: Region2::S3_inS2>: Execution of entry-activity started. State is now active.
[11] DEBUG - State <OwnedStateMachine1::Region1:: S2::Region2::S3_inS2>: Execution of entry-activity finished.
[12] DEBUG - Transition from Fork <OwnedStateMachine1 ::Region1{::UnNamedVertex}> to State <OwnedStateMachine1:: Region1::S2::Region1::S1_inS2>: Traversing started.
[12] INFO  - State <OwnedStateMachine1::Region1::S2:: Region1::S1_inS2>: Execution of entry-activity started. State is now active.
        (...)  //part omitted
[3] DEBUG - State diagram <OwnedStateMachine1>: Completion event <> generated by State <OwnedStateMachine1 ::Region1::S2::Region2::S3_inS2> has been dispatched.
[3] DEBUG - State diagram <OwnedStateMachine1>: Completion event <> generated by State <OwnedStateMachine1 ::Region1::S2::Region1::S1_inS2> has been dispatched.
[3] DEBUG - State diagram <OwnedStateMachine1>: Call-event <suspend_operA [ID=1]>. has been dispatched.
[16] DEBUG - State <OwnedStateMachine1::Region1:: S2::Region1::S1_inS2>: Execution of exit-activity started.
[16] INFO  - State <OwnedStateMachine1::Region1:: S2::Region1::S1_inS2>: Execution of exit-activity finished. State is now inactive.
[17] DEBUG - Transition from State <OwnedStateMachine1 ::Region1::S2::Region1::S1_inS2> to State <OwnedStateMachine1 ::Region1::S2::Region1:: S2_inS2>: Traversing started.
[17] INFO  - State <OwnedStateMachine1::Region1::S2:: Region1::S2_inS2>: Execution of entry-activity started. State is now active.
[17] DEBUG - State <OwnedStateMachine1::Region1 ::S2::Region1::S2_inS2>: Execution of entry-activity finished.
[18] INFO  - State <OwnedStateMachine1::Region1::S2 ::Region1::S2_inS2>: Execution of do-activity started.
[3] DEBUG - State diagram <OwnedStateMachine1>: Call-event <finish_operA [ID=2]>. has been dispatched.
        (...)  //part omitted
[22] INFO  - Join <OwnedStateMachine1::Region1 {::UnNamedVertex}> : Entered.
[22] DEBUG - Transition from Join <OwnedStateMachine1:: Region1{::UnNamedVertex}> to Final state <OwnedStateMachine1::Region1{::UnNamedVertex}>: Traversing started.
[22] INFO  - Final state <OwnedStateMachine1::Region1 {::UnNamedVertex}>: Entered.
[22] WARN  - State diagram <OwnedStateMachine1>: Exiting.
```

## References

[1] R. France, B. Rumpe, "Model-driven Development of Complex Software: A Research Roadmap" in Future of Software Engineering at ICSE'07, IEEE Soc., 2007, pp. 37-54.

[2] OMG Unified Modeling Language Superstructure v. 2.1.2, OMG Document formal/2007-11-02, 2007, http://www.uml.org

[3] MDA Guide, Ver. 1.0.1, Object Management Group Document omg/ 2003-06-01, 2003.

[4] S. Frankel, Model Driven Architecture: Appling MDA to enterprise computing. Wiley Press, Hoboken, NJ, 2003.

[5] A. Baruzzo, M. Comini, "Static verification of UML model consistency", *Proc. of the 3rd Workshop on Model DEvelopment, Validation and Verification, co-located. at MoDELS'06,* Genoa, Italy, 2006, pp. 111-126.

[6] C. Lange, M. R. V. Chaudron, J. Muskens, L. J. Somers and H. M. Dortmans, "An empirical investigation in quantifying inconsis-

tency and incompleteness of UML designs", in Proc. of $2^{nd}$ *Workshop on Consistency Problems in UML-based Software Development co-located atUML'03 Conf.* , San Francisko, USA, Oct 2003, pp. 26-34.

[7] R. Pilitowski, A. Derezinska, "Code Generation and Execution Framework for UML 2.0 Classes and State Machines", in: T. Sobh (eds.) Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, Springer, 2007, pp. 421-427.

[8] Derezinska, R. Pilitowski, "Event Processing in Code Generation and Execution Framework of UML State Machines", in L. Madeyski, M. Ochodek, D. Weiss, J. Zendulka (eds.) Software Engineering in progress, Nakom, Poznań, 2007, pp.80-92.

[9] Harel, H. Kugler, "The Rhapsody Semantics of Statecharts (or On the Executable Core of the UML)" (preliminary version), in SoftSpez Final Report, LNCS, vol. 3147, Springer, Heidelberg, 2004, pp. 325-354.

[10] STL: UML 2 Semantics Project, References, Queen's University http://www.cs.queensu.ca/home/stl/internal/uml2/refs.htm

[11] M. Crane, J. Dingel, "UML vs. Classical vs. Rhapsody Statecharts: Not All Models are Created Equal", in: MoDELS/UML 2005, LNCS, vol. 3713, Springer, Heidelberg, 2005, pp. 97-112.

[12] Y. Jin, R. Esser and J. W. Janneck, "A Method for Describing the Syntax and Semantics of UML Statecharts", Software and System Modeling, vol. 3 no 2, Springer, 2004, pp. 150-163.

[13] H. Fecher, J. Schönborn, "UML 2.0 state machines: Complete formal semantics via core state machines", in FMICS and PDMC 2006, LNCS vol. 4346, Springer, Hildelberg, 2007, pp. 244-260.

[14] Chauvel, J-M. Jezequel, "Code Generation from UML Models with Semantic Variation Points", in MoDELS/UML 2005, LNCS, vol. 3713, Springer, Heidelberg 2005, pp. 97-112.

[15] A. Egyed, "Fixing inconsistencies in UML designs", in Proc. of 29th Intern. Conf. on Software Engineering, ICSE'07, IEEE Comp. Soc., 2007.

[16] S. Prochanow, R. von Hanxleden, "Statecharts development beyond WYSIWIG", in G. Engels et al. (Eds.) MODELS 2007, LNCS 4735, Springer, Berlin Heidelberg, 2007, pp. 635-649.

[17] Ha L-K., Kang B-W., Meta-Validation of UML Structural Diagrams and Behavioral Diagrams with Consistency Rules, Proc. of IEEE Pacific Rim Conf on Communications, Computers and Signal Processing, PACRIM,Vol. 2., 28-30 Aug. (2003) 679-683.

[18] F. J. Lange, M. R. V. Chaudron, "Defects in industrial UML models - a multiple case study", *Proc. of the 2nd Workshop on Quality in Modeling, co-located. at MoDELS'07*, Nashville, TN, USA, 2007, pp. 50-64.

[19] Maraee, M. Balaban, "Efficient decision of consistency in UML diagrams with constrained generalization sets", in ", *Proc. of the 1st Workshop on Quality in Modeling, co-located. at MoDELS'06*, Genoa, Italy, 2006, pp. 1-14.

[20] J. Wuest, SDMetrics - the UML design measurement tool, http://www.sdmetrics.com/manual?LORules.html

[21] Rhapsody, http://www.telelogic.com/ (2008)

[22] A. Niaz, J. Tanaka, "Mapping UML Statecharts into Java code", in Proc. of the IASTED Int. Conf. Software Engineering, 2004 , pp. 111-116.

[23] S. J. Mellor, M. J. Balcer, Executable UML a Foundation for Model-Driven Architecture, Addison-Wesley, 2002.

[24] K. Carter, iUMLite - xUML modeling tool, http://www.kc.com

[25] IBM Rational Software Architect, http://www-306.ibm.com/software/rational

[26] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design patterns: elements of reusable object-oriented software*, Boston Addison-Wesley, 1995.

[27] J. Liberty, Programming C#, O'Reilly Media, 2005.

[28] Acceleo project http://www.acceleo.org

[29] Booch, Metamodel of object-oriented modeling language, http://www.booch.com.architecture

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 525 – 534

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Designing new XML Based Multidimensional Messaging Interface for the new XWarehouse Architecture

Ahmed Bahaa Farid
Helwan University, Faculty of Computers and Information,
Cairo, Egypt
Email:
{Ahmed.Bahaa@gmail.com}

Prof.Dr.Ahmed Sharaf Aldin Ahmed
Helwan University, Faculty of Computers and Information,
Cairo, Egypt
Email:
{Profase2000@yahoo.com}

Prof.Dr. Yehia Mostafa Helmy
Helwan University, Faculty of Computers and Information,
Cairo, Egypt
Email:
{Ymhelmy@yahoo.com}

*Abstract*— **OLAP (Online Analysis Processing) applications have very special requirements to the underlying multidimensional data that differs significantly from other areas of application (e.g. the existence of highly structured dimensions). In addition, providing access and search among multiple, heterogeneous, distributed and autonomous data warehouses, especially web warehouses, has become one of the leading issues in data warehouse research and industry. This paper proposes a new message interface for a new platform independent data warehouse architecture that can deliver location, platform, and schema transparency for clients that access autonomous data warehouses. The new message interface uses XML in order to provide interoperable way to query and administrate federated data warehouses in addition to compose the multidimensional query result sets.**

## I. Introduction

### A. Background

SINCE its evolution XML, accompanied with its related technologies (XQuery, XPath, XSL,…etc), has been considered the main standardized technology for the data exchange over information networks [1], [2]. By the time, the XML usage has increased with many applications. Recently, XML has significantly influenced building databases [2]. XML data is generated by applications and it can be consumed by applications. It is not too hard to imagine that some data sources in the enterprise are repositories of XML data or that they are viewed as XML data independently on their inner implementation.

In this case we could try to build a new data warehouse (DW) architecture that uses XML as its base for designing the messaging interface for that new architecture. In [3] a new data warehouse architecture (that is called XWarehouse) has been introduced. This architecture proposes the idea of utilizing XML as well as other design ideas in order to be able to build a platform independent multi- federated DW. In [3] the need for XML based messaging interface that has been called `eXtensible multiDimensional XML XDXML` has been introduced. As being part of the XWarehouse architecture, XDXML purpose is to compose all request, and response messages between the XWarehouse clients and orchestration. This will gives the opportunity to establish a communication between DW clients and a DW server with no regard to the platform compatibility issues. Moreover, XDXML provides the multidimensional format needed for caching the Multi-federated DW data as well as the DW Metadata on the Warehouse orchestration server that is responsible of orchestrating the multiple federated incompatible data warehouse at the backend with the DW query requests by the XWarehouse clients. In consequence, a need for dimensional modeling of XML data is appearing through the introduction of the XDXML. This paper proposes the XML based multidimensional messaging interface (XDXML) that the proposed XWarehouse uses. The paper depicts XDXML design goals, its basic structure, its embedded support for multi dimensions, its new operators, and the active capabilities into it. This XML based interface has been implemented into a real world case study that has been already presented previously [3].

### B. Contribution

The contribution of this paper aims to propose a new Multidimensional messaging interface in order to interchange multidimensional data, schemas, queries, and other administrative commands over XWarehouse data warehouse architecture [3]. This interface will be called `eXtensible Multidimensional XML (XDXML)`. The remainder of this study is organized. Specifically; to describes the architecture by means of:

- Overview of the related research work
- The XDXML design objectives.
- The XDXML schema description.
- The XDXML active commands as well as predicates

### C. Outline

This paper is organized as follows. At the beginning this paper presents a literature review that has conducted a survey about the previous efforts that tried to tackle using XML in dimensional modeling [4], [5]. The section analyzes each of those efforts and tells what is/are the negative point(s) into each proposed effort. Afterwards, the paper presents the

This work was supported into its practical side with Johnson & Johnson – Medical Egypt

main design objectives that the XDXML tries to fulfill. Prior to that, the paper delves into the XDXML basic multidimensional schema. In order present more details about the XDXML; the paper presents part of a real life implementation that has been used in order to validate the XDXML applicability. This helps in depicting the last part of the paper that depicts the XDXML ability to not only to compose multidimensional data as well as queries but also to send some administrative commands too.

### *D.Literature Review*

The study has made a reviews for finding out the research efforts that has tackled the same problem domain, or part of it. For the time being, and as to the researcher knowledge, only few research efforts have been done regarding utilizing the XML [6], [7] in creating an architectural foundation for storing, administrating, and integrating data warehouse data (i.e. multidimensional data).

#### *1)The Common Warehouse Metamodel (CWM)*

The CWM is an Object Management Group (OMG) initiative. Its version 1.0 has been released in Feb. 2001. Through the CWM specifications[7], [8], OMG is targeting the creation of standard format for data warehouse Metadata based on a foundation Metamodel [4]. Based on the UML, CWM builds a complex model for describing data warehouse Metadata. The main goal of its specification is to create a standard interface to data warehouses that every vendor tool can access, e.g. OLAP tools [5], ETL tools etc. [8], [9], The specification concentrate on building a standard between vendor tools for data warehouse interchange based on certain XML format [9]. The data warehouse multidimensional data interchange is out of scope of this specification. The CWM XML package only contains XML based definitions for classes and associations that represent common data warehouse Metadata. Based on this, the CWM just targets to bridge the gaps between data warehouse tools in order to be able to work together but, it doesn't provide message interface for exchanging and querying data in an open environment [8] ,[10],[11], [12]. By other words, CWM is a data interchange format more than being a multidimensional message interface that includes action commands to take remote actions. This is left for the application level not the CWM itself.

#### *2) MetaCube-X XML Metadata Foundation*

The MetaCube-X is an XML instance of the Metacube concept, [11], [12]. While MetaCube is a conceptual multidimensional data model that some vendors are currently using (e.g. Informix, and Microstrategy), MetaCube-X seeks providing the user with a query mechanism for accessing information on the different web warehouses. This concept of MetaCube-X concentrate on defining an XML based schema for querying multidimensional data from different web warehouses [11]. Based on this MetaCube-X only concentrates on Metadata and doesn't take care of the data itself (the fact data as well as the dimensions)[6]. Moreover, when proposing the data MetaCube-X proposes it tightly coupled with its Metadata. By other words, it doesn't make separation between schema and the multi-dimensional data. The MetaCube-X whole contribution is only targeting querying activities. Similarly to CWM it doesn't utilize the web services technology to provide access to remote web warehouse, and OLAP systems. Finally it doesn't support a specific architecture to implement its model as a complete solution.

#### *3) XML for Analysis (XMLA)*

XMLA is an initiative that has been co-sponsored by Microsoft and Hyperion (SAS has joint them in April 2002). Its version 1.0 specification has been released in April 2001 [13], [14], [15]. This specification utilizes the popularity of web services in providing data warehouse users with SOAP and XML based access to remote OLAP systems. While proposing an appropriate architecture, XMLA is concerned with how to query multidimensional data as well as Metadata through the use of the md/XML interface [13]. By other words, XMLA along with md/XML is designed to retrieving data not for manipulating and recapitulating cubes [13][14],. Moreover, XMLA doesn't support querying the galaxy schema for data retrieval

#### *4) XCube*

XCube is a family of XML based document templates that aim to exchange data warehouse data (i.e. data cubes) over any kind of networks [17]. In spite of releasing a way to exchange data as well as format, XCube schema is dedicated for the purpose of querying the web warehouse data , not any other purpose (i.e. exploring cube facts and dimensions, and managing a web warehouse) [18]. Not only that but also, the XCube schema is complicated to the extent that it is hard to be processed. This means that it doesn't support multiple Hierarchies dimensions. Within the schema description there is no distinction between the dimension itself and its dimensional hierarchies. XCube main target is to supply a standard format for exchanging data warehouse data, not a complete architectural solution for managing, querying, and exchanging data. There is no support for utilizing XSD for Schema validation or XSLT for presentation layer flexibility. There is no concrete definition for an architecture that defines how to deploy this format, or where its parsers will be deployed. Subsequently, XCube doesn't discuss how could be a level of integration between federated data warehouses, or web warehouses. Finally, it is not concerned with granting a high level of access to the data warehouses through something like Thin OLAP (ThOLAP) that X-Warehouse.

#### *5) INRIA's GEMO Project*

GEMO is a three-years project that has been born from the merging of INRIA (Institut National De Rechercha En Informatique Et En Automatique). With Members of another multinational research group. This project depends heavily on the usage of XML related technologies in order to insure data exchange as well as management. One of the main application domains for this project is the data for data warehouses. The Gemo 2006 activity report that has been published by INRIA doesn't talk about tackling the idea of the integration between federated data warehouses [19]. In order to promote their project foundation, the members of INRIA has contributed in releasing many publications regarding there diversified research points.

#### *6) Concluding Remarks*

These are all the known scientific contributions regarding the study research point. As has been clarified each one of the five contributions has some weak points that is fulfilled in this paper's contribution. Based on the above review it could be seen that none of the depicted efforts has proposed a complete architecture that enables platform independent data warehouse architecture that can enable integrating multi-federated data warehouses together which could be accessed transparently. While not supported by some of the efforts depicted above, XDXML proposes this through its architecture (XWarehouse). In addition to that, XDXML supports querying Galaxy schema. Moreover, XDXML supports making remote actions over the remote cubes.

## II. XDXML Design Goals

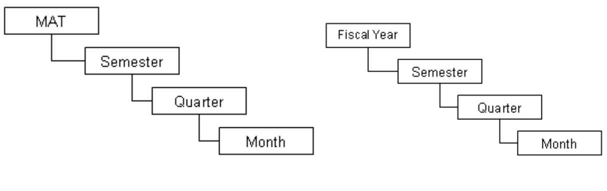
The XDXML format has been designed according to certain objectives. These objectives target functional as well as nonfunctional requirements. At the following are the main five design objectives that based on them the XDXML XML based Multidimensional Message Interface has been designed.

### *A. Minimizing Size*

At the core of the XDXML, the well formed XML resides. The main nature of any well formed XML document is that it is being constructed hierarchal. The hierarchal structure in turn imposes a larger size than other ways of composing documents (e.g. relational). According to that the XDXML should try to minimize the redundancies that may appear into the single document without scarifying any required functional requirement. Minimizing the size of the XDXML data and commands can enhance the performance of sending requests and receiving responses to and from the X-Warehouse server. In addition to that it can minimize the time needed to parse

### *B. Supporting Multiple Hierarchies*

In multidimensional design any cube can support multiple hierarchies [20], [21], [22]. This helps when same fact data is needed to get navigated with specific dimension according to multiple hierarchal points of view. As an example for that, for a pharmaceutical company they need to track the net sold amounts (after calculating all type of discounts) according to the time dimension. They have multiple time hierarchies Fig. 1 a, and b.



(a) Time Hierarchy 1 (b) Time Hierarchy 2

Fig. 1a: Time Dimension Hierarchy 1

The first hierarchy is concerned with dynamic periods. People at the pharmaceutical company have the need to track the sales values over the last 12 months (Moving Annual Time- MAT). This period is divided into two semesters (6 months). Each semester in turn is divided into 2 quarters (3 months) for each (see Fig. 1a). At the same time, people need to navigate the data for the current fiscal year, semester, and quarters. This brings up the second hierarchy (see Figure 1b). Incentives at the pharmaceutical companies are calculated based on each quarter sales and performance. Most of the work that has been proposed previously in this research field neglects delivering a schema that supports multiple hierarchies for the same dimension. This put them in a bad corner when it comes to the real implementation practices [18], [11].XDXML should deliver a multi-hierarchal dimension support.

### *C. Supporting Actions Declaration*

The XDXML doesn't target only the regular decision makers. By other words it doesn't only provide DW reports and queries requests and responses but, it targets supporting DW administrators too. That is why the Proposed X-Warehouse XDXML format supports two action commands that could be sent by administrators and other querying users to perform certain actions on the server. These commands are *Act* and *Get*

### *D. Supporting New Dimensional Operators*

XDXML should propose new dimensional operators. The new operators should deliver better querying capabilities. The new operators should help to get only data that is really needed.

### *E. Supporting Galaxy Schema*

The multi-fact cubes are not regular cubes but, sometimes the business needs impose having it. This may happen when extending the data warehouse schema with new measures that changes the core approach that the business work on. At this time, it is highly recommended to overcome the problems that may arise at this time. Moreover, the multi fact cube could be beneficial when targeting to build the data warehouse according to the Ralph Kimball's Bus architecture [21], [22]. Multi-Fact cube creates a multi-star schema that is called Galaxy Schema, or Fact Constellation schema [23]. In Galaxy Schema same dimensions are utilized by more than one fact table. XDXML X-Warehouse messaging interface should support the Galaxy Schema that is not supported into any of the previous work [24].

## III. The XDXML Schema

In order to fulfill the previous design objectives XDXML differentiates between the schema and the data itself. The schema could be explained separately through an XSD based file then the XDXML data comes accompanied with. At the following is an explanation of the XDXML schema, and how the data will get composed to.

### *A. The XDXML Cube*

The XDXML Cube is the key component of the XDXML schema. The cube is the container of the rest of the multidimensional data that will be gotten from the server. For better network performance, the XWarehouse architecture implements the idea of Cubes Client-Side Cashing

`(3Cs)` [3]. The 3Cs feature imposes transferring data into cube format not a report format. This can minimize the size of data transferred which fulfils the first design objective for the XDXML protocol which is Minimizing. Moreover, transferring the requested data as facts and dimensions can give the user more flexibility to make some slicing and dicing operations while not being connected to the server. According to that the *XDXML Cube* consists of *XDXML Facts,* and one or more *XDXML Dimensions.* Fig. 2 shows the main structure of the XDXML Cube.

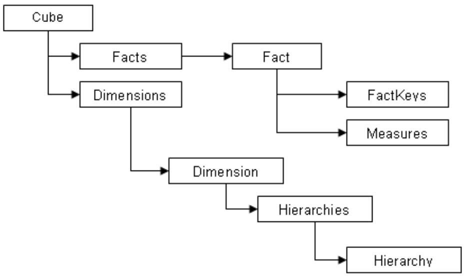


Fig. 2 The XDXML Cube main structural schema

As it is depicted each cube consists of one or many fact as well as one or many dimensions. As has been stated above the first design objective for the XDXML which is Minimizing Size has been fulfilled by transferring the multidimensional data not as cells but as facts and dimensions. This doesn't not only minimize the size of data received from the server but also, gives the flexibility to process the data on the client while being disconnect off the server. It is important to note here that each XDXML Cube node has an attribute called name for the cube name.

Fig. 3 Depicts the XDXML schema elements as classes using the UML based Class diagram.

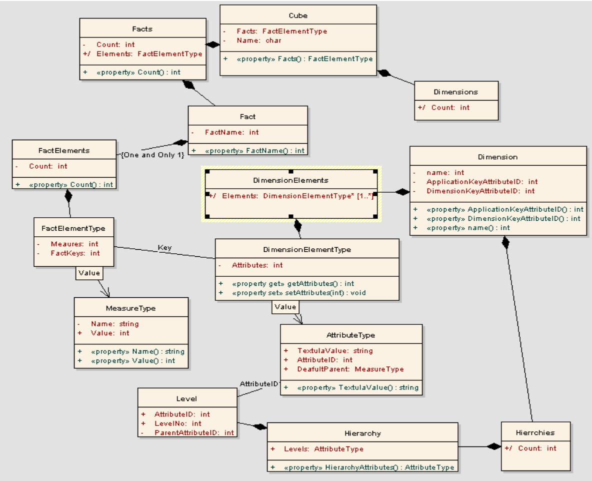
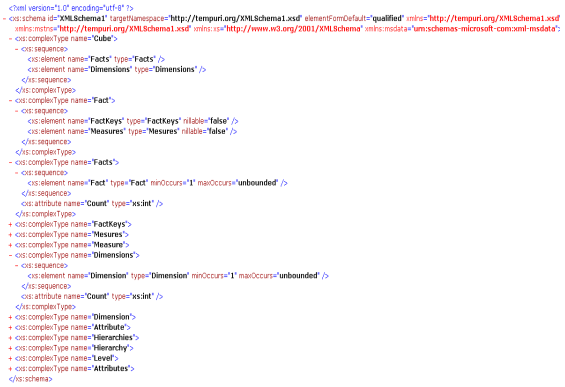

Facts
- Count: int
+/ Elements: FactElementType
+ «property» Count() : int
Cube
- Facts: FactElementType
- Name: char
+ «property» Facts() : FactElementType
Dimensions
+/ Count: int
Fact
- FactName: int
+ «property» FactName() : int
FactElements
- Count: int
+ «property» Count() : int
{One and Only 1}
DimensionElements
+/ Elements: DimensionElementType* [1..*]
Dimension
- name: int
- ApplicationKeyAttributeID: int
- DimensionKeyAttributeID: int
+ «property» ApplicationKeyAttributeID() : int
+ «property» DimensionKeyAttributeID() : int
+ «property» name() : int
FactElementType
- Meaures: int
- FactKeys: int
Key
Value
DimensionElementType
- Attributes: int
+ «property get» getAttributes() : int
+ «property set» setAttributes(int) : void
Value
MeasureType
- Name: string
+ Value: int
+ «property» Name() : string
+ «property» Value() : int
AttributeType
+ TextulaValue: string
+ AttributeID: int
+ DeafultParent: MeasureType
+ «property» TextulaValue() : string
AttributeID
Level
+ AttributeID: int
+ LevelNo: int
- ParentAttributeID: int
Hierarchy
+ Levels: AttributeType
+ «property» HierarchyAttributes() : AttributeType
Hierrchies
+/ Count: int

Fig. 3 A UML Representation for the main XDXML schema

Fig. 4 depicts the XML representation of the XDXML cube.

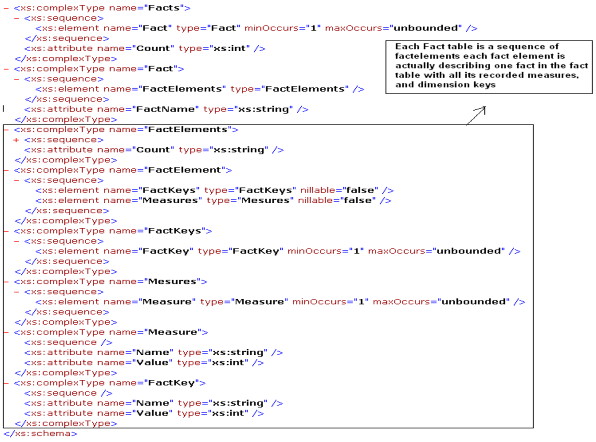


```
  <?xml version="1.0" encoding="utf-8" ?>
- <xs:schema id="XMLSchema1" targetNamespace="http://tempuri.org/XMLSchema1.xsd" elementFormDefault="qualified" xmlns="http://tempuri.org/XMLSchema1.xsd"
    xmlns:mstns="http://tempuri.org/XMLSchema1.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  - <xs:complexType name="Cube">
    - <xs:sequence>
        <xs:element name="Facts" type="Facts" />
        <xs:element name="Dimensions" type="Dimensions" />
      </xs:sequence>
    </xs:complexType>
  - <xs:complexType name="Fact">
    - <xs:sequence>
        <xs:element name="FactKeys" type="FactKeys" nillable="false" />
        <xs:element name="Measures" type="Mesures" nillable="false" />
      </xs:sequence>
    </xs:complexType>
  - <xs:complexType name="Facts">
    - <xs:sequence>
        <xs:element name="Fact" type="Fact" minOccurs="1" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="Count" type="xs:int" />
    </xs:complexType>
  + <xs:complexType name="FactKeys">
  + <xs:complexType name="Mesures">
  + <xs:complexType name="Measure">
  - <xs:complexType name="Dimensions">
    - <xs:sequence>
        <xs:element name="Dimension" type="Dimension" minOccurs="1" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="Count" type="xs:int" />
    </xs:complexType>
  + <xs:complexType name="Dimension">
  + <xs:complexType name="Attribute">
  + <xs:complexType name="Hierarchies">
  + <xs:complexType name="Hierarchy">
  + <xs:complexType name="Level">
  + <xs:complexType name="Attributes">
  </xs:schema>
```

Fig. 4 the XDXML Cube Schema

### *B. The XDXML Fact*

`XDXML Fact` is the heart of the cube. At most cases, cubes have just one fact table but, at some case their maybe more than one fact table. This happens in what is called galaxy schema. XDXML format supports both cases. Based on that XDXML schema may contain one or many cubes within the `XDXML Facts`. The `Facts` contains the sequence of `XDXML Fact`. Each Fact represents a fact table. Actually, this fulfils the fifth design objective for the XDXML protocol which is; Supporting Galaxy Schema. Each Fact consists of an attribute `FactName` and a Collection of `FactElements`. This is a of sequence of `FactElement` that represent one fact along with its measures and keys. That is why, each `FactElement` consists of two children elements; `Measures` element, and `Factkeys` element. The `Factkeys` element contains inside it all XDXML `FactKey` nodes. Each FactKey element node describes one of the FactKeys that connect the facts to their related dimensions. Each `FactKey` has two describing attributes. The `Measures` element contains a sequence of `Measure` nodes. Each Measure element has two attributes. Fig. 5 shows a fragment of the XDXML Cube schema that describes the XDXML Fact.



```
- <xs:complexType name="Facts">
  - <xs:sequence>
      <xs:element name="Fact" type="Fact" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="Count" type="xs:int" />
  </xs:complexType>
- <xs:complexType name="Fact">
  - <xs:sequence>
      <xs:element name="FactElements" type="FactElements" />
    </xs:sequence>
    <xs:attribute name="FactName" type="xs:string" />
  </xs:complexType>
- <xs:complexType name="FactElements">
  + <xs:sequence>
    <xs:attribute name="Count" type="xs:string" />
  </xs:complexType>
- <xs:complexType name="FactElement">
  - <xs:sequence>
      <xs:element name="FactKeys" type="FactKeys" nillable="false" />
      <xs:element name="Measures" type="Mesures" nillable="false" />
    </xs:sequence>
  </xs:complexType>
- <xs:complexType name="FactKeys">
  - <xs:sequence>
      <xs:element name="FactKey" type="FactKey" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <xs:complexType name="Mesures">
  - <xs:sequence>
      <xs:element name="Measure" type="Measure" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <xs:complexType name="Measure">
    <xs:sequence />
    <xs:attribute name="Name" type="xs:string" />
    <xs:attribute name="Value" type="xs:int" />
  </xs:complexType>
- <xs:complexType name="FactKey">
    <xs:sequence />
    <xs:attribute name="Name" type="xs:string" />
    <xs:attribute name="Value" type="xs:int" />
  </xs:complexType>
</xs:schema>
```

Each Fact table is a sequence of factelements each fact element is actually describing one fact in the fact table with all its recorded measures, and dimension keys

Figure 5: The XDXML Fact Schema

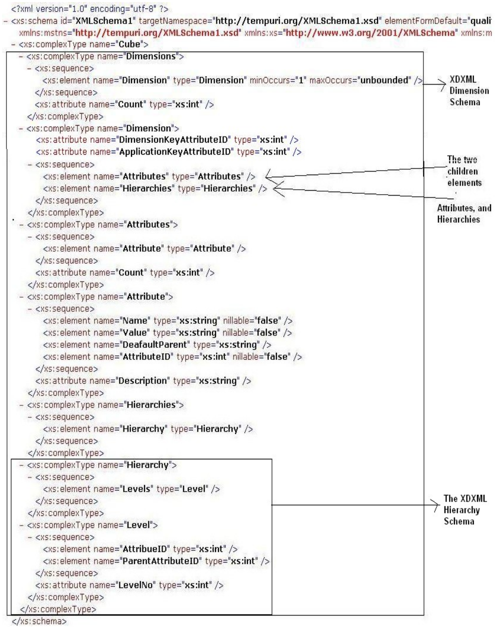
### *C. The XDXML Dimension*

The Dimension element has a count attribute. This helps in defining explicitly how many dimensions is contained into any XDXML document that can enhance the parsing algorithm performance. Each XDXML Dimension element describes one dimension table. The Dimension element has two children, and three attributes. The first Attributes is the Name attribute (This attributes is not presented into figure 6 because of the space limitation), the second, and third attributes are attributes for declaring the Dimension Key, and the Application Key. These are DimensionKeyAttributeID, and ApplicationKeyAttributeID attributes respectively.

The first child is the `Attributes` element of `Attributes` type. The Attributes type is a sequence of XDXML Attribute element that compose up the dimension table. Each `Attribute` element usage purpose is depicted into a XML attribute called Description describing what this is used for. In addition to that each attributes has four describing nodes. The `Name` node element states the attribute name. The `Value` node contains the attribute value. The `AttributeID` node grants each attribute an integer id. This ID helps in referring to each attribute for many purposes that will appear later. Using integer ID instead of the string attribute id (attribute name) to refer for the XDXML attribute can enhance the parsing performance as well as minimizing the XDXML size. It is important here to state that the Dimension Key as well as the Dimension Application Key [21], [22] are included as two XML attribute elements. The values of these two attributes refer to the values of the AttributeID element node value within the two XDXML Attributes that act as Dimension ID and Application ID. The fourth attribute is The `Default Parent` node element is the AttributeID of the upper level for each attribute in the default hierarchy. Fig. 6 shows the XDXML Dimension Schema. The second child for *the XDXML Dimension* element node is the Hierarchies of a Hierarchies type. The Hierarchies Type contains a sequence of *XDXML Hierarchy* element nodes.

### *D. The XDXML Hierarchy*

The `XDXML Hierarchy` is no more than a definition for how data will get organized. It has no more user data than that already exists into the dimension attributes. Instead, it organizes the existing attributes by specific organization only into certain granularity levels. According to that, the `XDXML Hierarchy` is a part of the `XDXML Dimension` (see Fig. 6). As it is apparent at the figure, the XDXML Hierarchy schema contains one direct child node; the `Levels` element node of Levels type. In turn, the Levels element is a sequence of Level element nodes type. Each Level element node has one attribute that describe the Level number of each level in the hierarchy. The description of the attribute that resides at each level comes for two element nodes; the AttributeID and the ParentAttributeID that contains the AttributeID value for the attribute above it in the Hierarchy. This approach of describing the multi-hierarchal schema in the XDXML schema makes it distinctive of the other related work that talked about the multi-hierarchal schema because; it makes the Multi-hierarchal description on the schema itself not on the data, as others do [7],[9], [11], [18],. Again, this can minimize the size aggressively. The XDXML Hierarchy Schema fulfills the second design objective for that XDXML that has been stated above which is: *Supporting Multiple Hierarchies.*



```
  <?xml version="1.0" encoding="utf-8" ?>
- <xs:schema id="XMLSchema1" targetNamespace="http://tempuri.org/XMLSchema1.xsd" elementFormDefault="quali
    xmlns:mstns="http://tempuri.org/XMLSchema1.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:m
  - <xs:complexType name="Cube">
    - <xs:complexType name="Dimensions">
      - <xs:sequence>
          <xs:element name="Dimension" type="Dimension" minOccurs="1" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:attribute name="Count" type="xs:int" />
      </xs:complexType>
    - <xs:complexType name="Dimension">
        <xs:attribute name="DimensionKeyAttributeID" type="xs:int" />
        <xs:attribute name="ApplicationKeyAttributeID" type="xs:int" />
      - <xs:sequence>
          <xs:element name="Attributes" type="Attributes" />
          <xs:element name="Hierarchies" type="Hierarchies" />
        </xs:sequence>
      </xs:complexType>
    - <xs:complexType name="Attributes">
      - <xs:sequence>
          <xs:element name="Attribute" type="Attribute" />
        </xs:sequence>
        <xs:attribute name="Count" type="xs:int" />
      </xs:complexType>
    - <xs:complexType name="Attribute">
      - <xs:sequence>
          <xs:element name="Name" type="xs:string" nillable="false" />
          <xs:element name="Value" type="xs:string" nillable="false" />
          <xs:element name="DeafaultParent" type="xs:string" />
          <xs:element name="AttributeID" type="xs:int" nillable="false" />
        </xs:sequence>
        <xs:attribute name="Description" type="xs:string" />
      </xs:complexType>
    - <xs:complexType name="Hierarchies">
      - <xs:sequence>
          <xs:element name="Hierarchy" type="Hierarchy" />
        </xs:sequence>
      </xs:complexType>
    - <xs:complexType name="Hierarchy">
      - <xs:sequence>
          <xs:element name="Levels" type="Level" />
        </xs:sequence>
      </xs:complexType>
    - <xs:complexType name="Level">
      - <xs:sequence>
          <xs:element name="AttribueID" type="xs:int" />
          <xs:element name="ParentAttributeID" type="xs:int" />
        </xs:sequence>
        <xs:attribute name="LevelNo" type="xs:int" />
      </xs:complexType>
    </xs:complexType>
  </xs:schema>
```

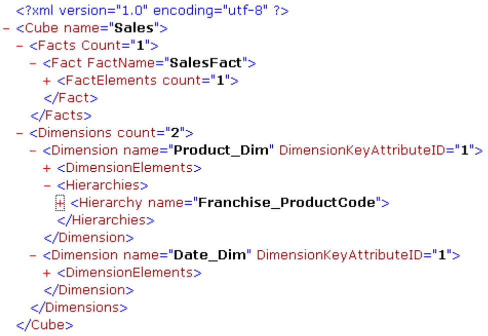
XDXML Dimension Schema
The two children elements
Attributes, and Hierarchies
The XDXML Hierarchy Schema

Figure 6: The XDXML Dimension Schema

## IV. Applying the XDXML Schema on a Real Multidimensional Data

In order to validate this architecture, a case study has been conducted in one of the Medical equipment multinational companies in Egypt (Johnson & Johnson-J&J- medical Egypt), the architectural components have been belt using .NET C# code and then got deployed. The company hosts two data marts. The first one is the In Market Sales data marts (this includes facts about selling the devices and items from the distributor to the end-user) that keep the sales team distribution, Sales budgets and sales actual achievements. This data mart is about 2.54 GB in size. The Financial data mart is a 3.2 GB data mart that is hosted on an AS/400 server that is located on Europe (this includes the financial facts as well as the facts of sales between the company and distributors). This data mart is related to the internal sales between the Egyptian subsidiary and the EMEA headquarter. Previously all strategic planning used to be done on the numbers of the To Market sales. According to some deficiencies that has been discovered during the last two years, the corporate has changed its global strategy to make both the To-market Sales, and the In-Market Sales equally important. Based on this, new measures have been emerged based on the new interest of having the ability to process the internal To-Market sales(J&J/ Distributor) along with the In-Market Sales(distributor/end customer) . That is why a new need has been emerged to navigate to both of measurements together. All used examples below are extracted out of this case study as a try to highlight practical examples for using the

XDXML in exchanging a subset of a multidimensional data of a multinational medical company. This multinational medical company is specialized in producing and selling the medical materials (bondage, medical threads, plasters…). Fig. 7 depicts an overview of the multidimensional data that is retrieved from the orchestration server of this company. As it is clear from the figure, the cube is composed of one Fact called *SalesFact* and two dimension elements. The first one is the *Product_Dim* dimension while the second is the *Date_Dim*.

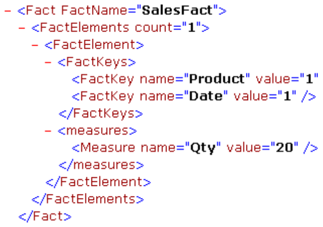


```
  <?xml version="1.0" encoding="utf-8" ?>
- <Cube name="Sales">
  - <Facts Count="1">
    - <Fact FactName="SalesFact">
      + <FactElements count="1">
      </Fact>
    </Facts>
  - <Dimensions count="2">
    - <Dimension name="Product_Dim" DimensionKeyAttributeID="1">
      + <DimensionElements>
      - <Hierarchies>
        + <Hierarchy name="Franchise_ProductCode">
        </Hierarchies>
      </Dimension>
    - <Dimension name="Date_Dim" DimensionKeyAttributeID="1">
      + <DimensionElements>
      </Dimension>
    </Dimensions>
  </Cube>
```

Fig. 7 Practical Example for a XDXML Cube

### *A. The SalesFact XDXML Fact*

For the sake of simplicity this fact has only one measure which is the *Qty*. Actually this depicts the quantity sold from the referred product at the referred time. These three pieces of data compose all together one fact that is recorded as one `FactElement`. Fig. 8 depicts the `SalesFact` XDXML fact. As it is opposed, This XDXML fact has just one fact element (this done to simplify the example) The fact element shows up that this XDXML has two keys that refer to two dimensions (remember that the cube has two dimensions that are referred to by these two keys). The first `FactKey` is the *Product* key that refers to the dimension key number 1 in the `Product_Dim` dimension. The second key is the Date key that refers to the dimension key number 1 in the *Date* dimension.

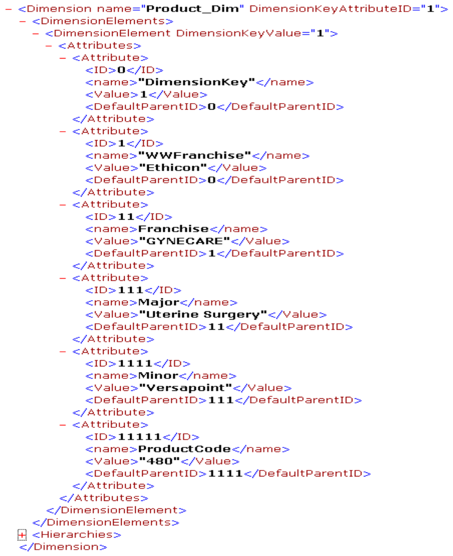


```
- <Fact FactName="SalesFact">
  - <FactElements count="1">
    - <FactElement>
      - <FactKeys>
          <FactKey name="Product" value="1" />
          <FactKey name="Date" value="1" />
        </FactKeys>
      - <measures>
          <Measure name="Qty" value="20" />
        </measures>
      </FactElement>
    </FactElements>
  </Fact>
```

Figure 8 The *SalesFact* XDXML Fact

### *B. Product_Dim XDXML Dimension*

Fig. 9 depicts the Product_Dim XDXML dimension. As it is apparent from the figure, this dimension has one default hierarchy and another defined hierarchy. For the sake of simplicity just one dimension element has been defined here. As it is clear from the figure the only dimension element available has a dimension key with value 1. This is the same value for the fact key at the SalesFact XDXML fact. It is important here to highlight that the DefaultParentID element is used to represent the default hierarchy.



```
- <Dimension name="Product_Dim" DimensionKeyAttributeID="1">
  - <DimensionElements>
    - <DimensionElement DimensionKeyValue="1">
      - <Attributes>
        - <Attribute>
            <ID>0</ID>
            <name>"DimensionKey"</name>
            <Value>1</Value>
            <DefaultParentID>0</DefaultParentID>
          </Attribute>
        - <Attribute>
            <ID>1</ID>
            <name>"WWFranchise"</name>
            <Value>"Ethicon"</Value>
            <DefaultParentID>0</DefaultParentID>
          </Attribute>
        - <Attribute>
            <ID>11</ID>
            <name>Franchise</name>
            <Value>"GYNECARE"</Value>
            <DefaultParentID>1</DefaultParentID>
          </Attribute>
        - <Attribute>
            <ID>111</ID>
            <name>Major</name>
            <Value>"Uterine Surgery"</Value>
            <DefaultParentID>11</DefaultParentID>
          </Attribute>
        - <Attribute>
            <ID>1111</ID>
            <name>Minor</name>
            <Value>"Versapoint"</Value>
            <DefaultParentID>111</DefaultParentID>
          </Attribute>
        - <Attribute>
            <ID>11111</ID>
            <name>ProductCode</name>
            <Value>"480"</Value>
            <DefaultParentID>1111</DefaultParentID>
          </Attribute>
        </Attributes>
      </DimensionElement>
    </DimensionElements>
  + <Hierarchies>
  </Dimension>
```

Fig. 9 The Product_Dim XDXML Dimension

## V. Establishing Actions Using XDMQueries

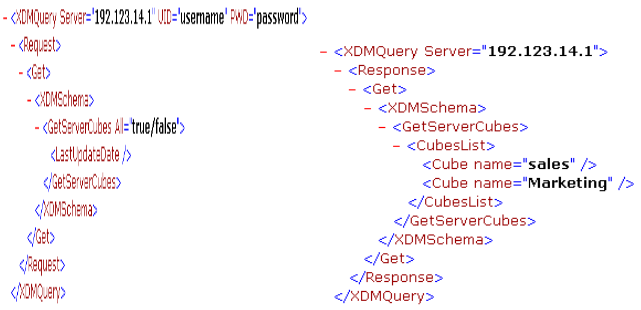
The third XDXML design objective imposes Supporting Actions Declaration. In order to fulfill this objective XDXML has introduced a way to query and administer the XWarehouse. This is done through `XDMQueries`**.** Actually, XDMQuery is part of the XDXML. This type of queries is sent within a XDMQuery XDXML node that can include *Get* and/or *Act* statements. The Get statement is primarily concerned with querying data and schemas, while the Act statement is concerned with performing some administrative actions on the remote cubes. Each query is sent within a request. The request may contain *Get* or *Act* statements. The answer is received within a XDMQuery Response tag.

### *A. The XDMQuery Get Statement*

Using the `XDMQuery` Get command it is possible to explore data as well as schema. This is done through sending an XDMQuery Request and receiving the response using the XDXML protocol for the schema or/and the data. Get statement works by using some XDMQuery predicates. The first predicate is <GetServerCubes>. If the request is containing a query with a response containing a schema(Metadata) then the statement should contain the predicates inside a `XDMSchema` tag. If the query needs to respond by data, not a schema then, the query predicates should be contained inside a `XDMData` tag. As depicted as Fig.10 the `XDMSchema` and `XDMData` tags are containing the predicates. This helps in containing multiple predicates at the same request some to receive data and others to receive Metadata.

#### *1) GetServerCubes Predicate(Schema only)*

Fig.10 depicts the first XDMQuery Get statement predicate; the `GetServerCubes` predicate. This predicate is responsible of acquiring the server cubes available at specific server. This could be unlimited to all available cubes on the server or limited to the cubes last updated at specific time.



```
- <XDMQuery Server="192.123.14.1" UID="username" PWD="password">
  - <Request>
    - <Get>
      - <XDMSchema>
        - <GetServerCubes All="true/false">
            <LastUpdateDate />
          </GetServerCubes>
        </XDMSchema>
      </Get>
    </Request>
  </XDMQuery>
```

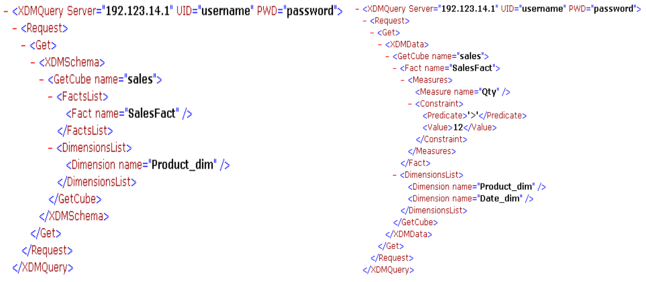
```
- <XDMQuery Server="192.123.14.1">
  - <Response>
    - <Get>
      - <XDMSchema>
        - <GetServerCubes>
          - <CubesList>
              <Cube name="sales" />
              <Cube name="Marketing" />
            </CubesList>
          </GetServerCubes>
        </XDMSchema>
      </Get>
    </Response>
  </XDMQuery>
```

(a) GetServerCubes Predicate Query (b) Query Response

Fig. 10 The GetServerCubes Request and Response in Get Statement

### 2) GetCube Predicate(Schema and data)

This predicate targets querying both; part of the specific cube data and/or its Metadata. As explained before, in order to define whether you need the schema for the required query or the Metadata you Should use either the XDMSchema or the XDMData tags. Fig. 11(a) shows how to use the GetCube predicate within the Get Statement in order to query the Cube Metadata. Fig. 11(b) shows an Example for the GetCube predicate when is used to retrieve data. If nothing else has been specified, the default is to return the data with the most granular dimensional level. The example that is depicted below doesn't mention any granularity levels for the Product_Dim dimension. If nothing else has been mentioned, the default hierarchy is retrieved, and the most granular level is used. If nothing else has been specified, the default is to return the data with the most granular dimensional level. The example that is depicted above doesn't mention any granularity levels for the Product_Dim dimension. If nothing else has been mentioned, the default hierarchy is retrieved, and the most granular level is used.



```
- <XDMQuery Server="192.123.14.1" UID="username" PWD="password">
  - <Request>
    - <Get>
      - <XDMSchema>
        - <GetCube name="sales">
          - <FactsList>
              <Fact name="SalesFact" />
            </FactsList>
          - <DimensionsList>
              <Dimension name="Product_dim" />
            </DimensionsList>
          </GetCube>
        </XDMSchema>
      </Get>
    </Request>
  </XDMQuery>
```

```
- <XDMQuery Server="192.123.14.1" UID="username" PWD="password">
  - <Request>
    - <Get>
      - <XDMData>
        - <GetCube name="sales">
          - <Fact name="SalesFact">
            - <Measures>
                <Measure name="Qty" />
              - <Constraint>
                  <Predicate>'>'</Predicate>
                  <Value>12</Value>
                </Constraint>
              </Measures>
            </Fact>
          - <DimensionsList>
              <Dimension name="Product_dim" />
              <Dimension name="Date_dim" />
            </DimensionsList>
          </GetCube>
        </XDMData>
      </Get>
    </Request>
  </XDMQuery>
```

(a)Retrieving Schema (b) Retrieving Data

Figure 11 GetCube Predicate Request

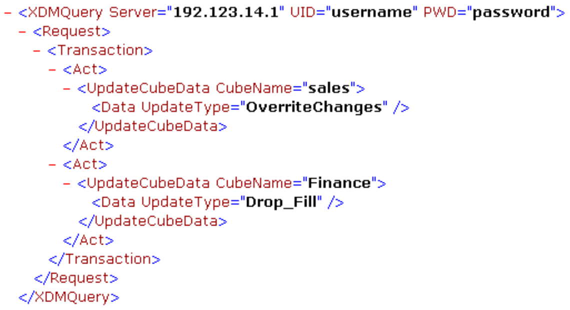
### B. The XDMQuery Act Statement

The *Act* is primarily directed to the administrators. By using the `Act` statement, it is possible to take action over existing cubes. One of the most important activities that the administrator may need to accomplish is to update his cubes. The `UpdateCubeData` is a predicate that could be used in order to initiate this task by the administrator remotely. For the time and resources limitation, this study presents only this predicate with the `Act` statement. More predicates could be provided for both the `Get` and the `Act` statements during future work.

Fig.12 depicts an example for the Act statement with the UpdateCubeData operator. As it is clear from the figure, the Act statement is enclosed inside a transaction. This to tell that all the statements contained inside the transaction elements are in just one transaction. The XDMQuery transaction can contain Act as well as Get statements. Enclosing Get statement inside a Transaction maybe helpful because that the XDMQuery may work in asynchronous mode. Based on this enclosing the Get statement inside a transaction will be a declaration that all enclosed statements will work as one batch. If one fails all will fail too. The Transaction purpose is clearer in Act statement. If one fails all actions taken on the cubes will rollback.

The *UpdateCubeData* predicate has one attribute to define the cube name. In addition to that it has one child element to define how data will be updated inside the cube. Whether data will be completely deleted and added again, or just the modified data will be overwritten. The former could be used if the cube data has changed massively at the base data warehouse since last update while, the later could be useful if there is no massive change in the data warehouse data.

Separating the update of each cube in separate Act statement rather than having Cubes child element will give the opportunity to make one cube got updated when others fail to perform their updates. If *All or Nothing* is needed it is possible to use the *XDMQuery Transaction* processing is needed. It is important the mention here that the cube updating alternative is supported if on ly the server side cubes management system supports this feature otherwise; the default cube updating method will take place.



```
- <XDMQuery Server="192.123.14.1" UID="username" PWD="password">
  - <Request>
    - <Transaction>
      - <Act>
        - <UpdateCubeData CubeName="sales">
            <Data UpdateType="OverriteChanges" />
          </UpdateCubeData>
        </Act>
      - <Act>
        - <UpdateCubeData CubeName="Finance">
            <Data UpdateType="Drop_Fill" />
          </UpdateCubeData>
        </Act>
      </Transaction>
    </Request>
  </XDMQuery>
```

Fig. 12 a Sample for the XDMQuery Act Statement

### C. Presenting the Pump-Up and Dump-down new Multidimensional Operators for Get command

The *XDQuery* supports two new operators for better multidimensional querying. These operators are directed to minimize the hierarchy declarations. Using these operators, the user can ask for a retrieving the data based on certain schema but without having all its intermediate levels. As an example for this, the hierarchy shown in figure 10(a) is depicting the default dimension hierarchy which is; WWFranchise/Franchise/Major/Minor/ProductCode. The question her is, what if the decision maker needs to retrieve the data according to the following Hierarchy; WWFranchise/Franchise/ProductCode? In order to fulfill this requirement, it is needed to define a new hierarchy with this structure. The new XDQuery operators help at this case. The `DumpDown` operator is responsible of aggregating data to one of the indirect parent of specific level. This means that when using the `PumpUp` operator, the query should define the most granular level and which parent level is direct aggregating the data. By other words using the DumpDown and PumpUp operators, the same required results could be gotten but DumpDown will retrieve some extra data that are not retrieved by PumpUp operator. This is means that using the same facts/dimensions/ level names Dump-Down data set is s upper se t of the PumpUp data set. The other difference is the hierarchy level that is used to refer to the data; whether it is an upper level or lower one. Figure 13 shows a sample hierarchy of the case study in subject of this study. This hierarchy is based on the same data of the previous figures.

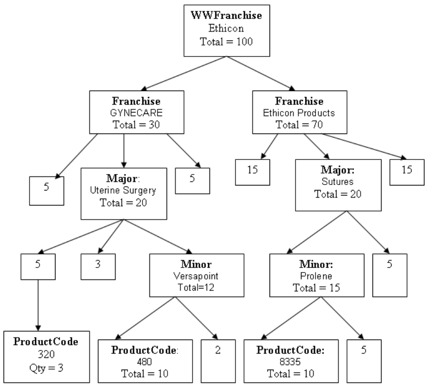


Fig. 13 Sample Data Hierarchy for the XDXML Cube Presented in figure 7 and all its subsequent figures

### *1) The Pump-Up XDMQuery Operator*

Fig.14 depicts syntax for using the PumpUp operator. The *Get* is the *XDMQuery's Get statement*. *With* is a reserved word for defining specific dimensions. *For* is another reserved word for defining specific value for the required hierarchy level.

***Get*** Sales**:** SalesFact(Qty)
***With** Product-Dim*
**On** ProductCode ***PumpUp*** Franchise
***For*** ProductCode = 8335

Fig 14 an example for Using PumpUp XDQuery Operator

According to figure 15, the result for this query will be as depicted in figure 16 but using XDXML. Figure 17 shows the same result but in XDXML The importance of using DumpDown and DumpUp operators comes from the fact that they minimize the amount of data that could be retrieved because it gives the ability to omit the unwanted intermediate levels of dimensional levels

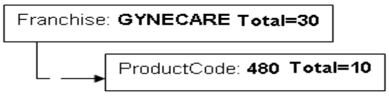
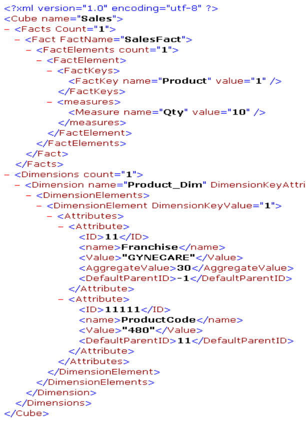


Fig. 15 Block Diagram for the Result of Query Shown in Fig.14



```
  <?xml version="1.0" encoding="utf-8" ?>
- <Cube name="Sales">
  - <Facts Count="1">
    - <Fact FactName="SalesFact">
      - <FactElements count="1">
        - <FactElement>
          - <FactKeys>
              <FactKey name="Product" value="1" />
            </FactKeys>
          - <measures>
              <Measure name="Qty" value="10" />
            </measures>
          </FactElement>
        </FactElements>
      </Fact>
    </Facts>
  - <Dimensions count="1">
    - <Dimension name="Product_Dim" DimensionKeyAttributeID="1">
      - <DimensionElements>
        - <DimensionElement DimensionKeyValue="1">
          - <Attributes>
            - <Attribute>
                <ID>11</ID>
                <name>Franchise</name>
                <Value>"GYNECARE"</Value>
                <AggregateValue>30</AggregateValue>
                <DefaultParentID>-1</DefaultParentID>
              </Attribute>
            - <Attribute>
                <ID>11111</ID>
                <name>ProductCode</name>
                <Value>"480"</Value>
                <DefaultParentID>11</DefaultParentID>
              </Attribute>
            </Attributes>
          </DimensionElement>
        </DimensionElements>
      </Dimension>
    </Dimensions>
  </Cube>
```

Fig 16 The XDXML Cube of the Get Statement at Fig 15

### *2) Dump-Down XDMQuery Format*

Fig. 17 shows the *DumpDown* operator. As presented, the *DumpDown* operator is a way that can be used to omit a lot of unneeded data. Fig.17 expresses that the user needs only to get the Qty Measure that is inside the SalesFact using the Franchise Level and its children up to the Product Code directly at the Product_Dim dimension. The result for this query is depicted in Fig. 18. As it is clear the main difference between *DumpDown* and *PumpUp* operators is that, *PumpUp* operator is used to get certain upper level data along with its all children at the dimensional hierarchy, while the *PumpUp* operator retrieves certain measurement value according to its value for certain leaf node value at specific hierarchy along with its defined parent at certain level. According to that, *DumpDown* will most probably returns amount of data that is larger in size of that returned by the PumpUp if same level names have been used at both statements.

***Get*** Sales: SalesFact(Qty)
***With Product-Dim***
***On*** Franchise ***DumpDown*** ProductCode
***For*** Franchise = GYNECARE

Fig. 17 An example for Using DumpDown XDMQuery Operator

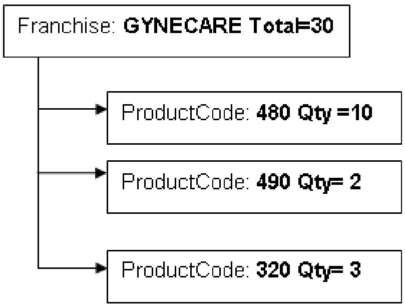


Fig.18 The XDXML Cube that represents the result of the Get Statement at Fig. 17

## VI. Future Work and Conclusion

### A. Future Work

This scientific work could be extended at the future by many ways. The following are just some examples of the potential future work:

-Designing more predicates for the existing Get and Act commands. These predicates could be related to more administrative tasks (e.g. building and dropping cubes)

-Taking in considerations the security aspect of XDXML as this paper doesn't discuss that aspect.

-Implementing optimized parsing algorithm for parsing XDXML at the XWarehouse client-side as well as service-side.

-Extending XDMQuery in order to be able to query available mining models.

-Extending XDXML to include querying data mining models [23].

### B. Conclusion

This work presented a message Interface for exchanging data as well as commands over the XWarehouse Architecture [19] using XML as well as its related technologies (e.g. XSD, XSLT,…). The XWarehouse architecture is a newly proposed data warehousing architecture that uses Web Warehousing infrastructure for building better platform independent and more integrated and accessible data warehouses. Web warehousing refers to whether the use of the power of web infrastructure in architecting data warehouses, or the use of data warehouse techniques into keeping historical data about the click stream backlog [22]. In the case of XWarehouse first meaning is the intended and meant.

The paper used the XML technology as the base for building up the XDXML. XDXML used inside XWarehouse to get used by clients and servers for exchanging their data, schemas, and query as well as administrative commands. This paper is not concerned with the security details of the XDXML. XWarehouse uses http and SOA (Service Oriented Architecture) in order to build interoperable, vendor neutral data warehouse architecture that can enable clients as well as servers that depends on different platforms, DWMS (Data Warehouse Management Systems), and Operating systems to interoperate together transparently from the client. Moreover X-Warehouse architecture provides a way in order to integrate multi-federated data warehouse to act transparently as one logical data warehouse in front of the X-Warehouse clients.

This research has introduced the XDXML by providing a review for the related work that used XML for encapsulating data warehouse data and queries. Moreover, this paper tried to unveil the weak points for each effort that XDXML tries to overcome [23]. Moreover, the paper clarified what is the major design objectives that has been build based on. Presenting the design objectives has illustrated how does XDXML overcomes all the depicted weak points at the related research efforts surveyed. Prior to that, the paper began to delve into the detailed design of the XDXML by explain the key constructing components of the XDXML data schema. This paper has used the same case study at which the XWarehouse has been implemented in order to apply the XDXML. Part of the sample data of this case study has been used as an example for depicting examples of the XDXML Cube, XDXML Fact, and XDXML dimension schemas. All these schemas are mainly based on the XSD (XML Schema Definition). In addition to that the paper showed how XDXML can express action either for querying multidimensional data or to do some administrative actions at the server side through what has been called XDMQuery. XDMQuery is a dual-verb XML based language that is considered part of the XDXML. The two verbs are *Get* for querying data, and *Act* for sending action requests including administration actions. Get verb can use to newly proposed operators at this paper in order better retrieve what is needed accurately without any extra dimensional data member levels. Thus, reducing the retrieved the size of the XDXML which is one of its design goals.

It is important to mention here that the case study has been conducted to apply the X-Warehouse along with its XDXML messaging interface using a real life environment that is brought from the medical industry of one of the multinationals in Egypt In order to verify the applicability of the architecture [3]. The result showed that:

-The X-Warehouse Architecture is possible to get implemented and fulfilling its design goals.

Figure 19 depicts the deployed X-Warehouse architecture. As it is apparent, in addition to the regular desktop as well as web based access, this architecture permits mobile devices from accessing web warehouses data.

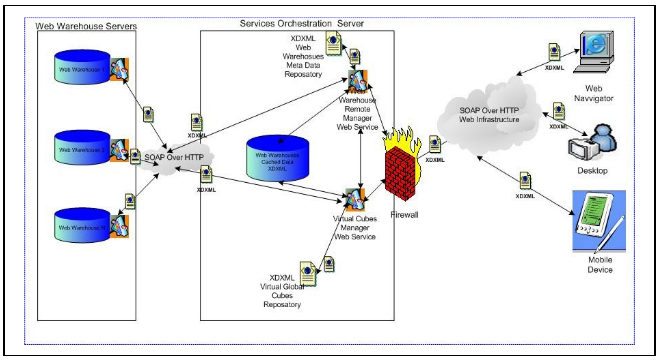

Web Warehouse Servers
Services Orchestration Server
Web Warehouse 1
Web Warehouse 2
Web Warehouse N
SOAP Over HTTP
XDXML
XDXML
XDXML Web Warehosues Meta Data Reposatory
Web Warehouse Remote Manager Web Service
Web Warehouses Cached Data XDXML
Virtual Cubes Manager Web Service
XDXML Virtual Global Cubes Reposatory
Firewall
XDXML
SOAP Over HTTP Web Infrastructure
XDXML
Web Navvigator
XDXML
Desktop
XDXML
Mobile Device

Fig 19 The X-Warehouse Architecture

-Using Caching in both server side as well as client side of the X-Warehouse architecture enhance the performance in average by 313%

-The performance overhead of using the X-Warehouse is in average just 5.11% which could be worthy to be incurred in order to solve the compelling problems that X-Warehouse tackles.

- The ThOLAP could be implemented based on the same architecture using the XDXML application protocol.

To conclude this work has successfully implemented the XDXML XML based XWarehouse Messaging Interface. A future studies will be made in order to more predicates to the Get and Act verbs for performing other administrative tasks. The XDXML at it is current state doesn't tackle how to query the data mining structures out from the server side.

This is expected to be the subject of one of the future research efforts.

## References

[1] Xiaogang Li and Gagan Agrawal, "*Efficient Evaluation of XML Over Streaming Data",* Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005.

[2] Wanhung Xu, Z. Meral Ozsoyoglu, *"Rewriting XPath Queries Using Materialized Views",* Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

[3] Ahmed Bahaa, Ahmed Sharaf, and Yahia Helmy, *"Towards a New Platform Independent Data Warehouse Architecture Using Web, and XML Technologies",* 2008

[4] Bert Scalzo, "*Oracle® DBA Guide to Data Warehousing and Star Schemas",* Prentice Hall. 2003.

[5] E. F. Codd, S. B. Codd, and C. T. Salley, "*Providing OLAP (Online Analytical Processing) to User Analysts: An IT Mandate",* White Paper, Arbor Software Corporation, 1993.

[6] Dalamagas Theodore, etal, *"A methodology for clustering XML documents by structure, Information Systems",* Information Systems Journal No. 31, 2006, Elsevier B.V. P.187 -228.

[7] Pokorny Jaroslav. "*Modeling Stars Using XML",* Proceedings of DOLAP'01 ACM. Atlanta, United States, November 9, 2001.

[8] Kumpon Farpinyo, "*Designing and Creating Relational Schemas with a CWM-Based Tool",* ACM digital library, 2003.

[9] N. N.: "*Common Warehouse Meta Model Specification. Version 1.0",*.OMG Feb. 2001. TTTT http://www.omg.org/cgi-bin/apps/doc?ad/01-02-01.pdf

[10] Claudio Seidman, "*Data Mining with Microsoft® SQL Server™ 2000 Technical Reference",* Microsoft. 09/2001.

[11] T. B. Nguyen, A M. Tjoa, R. R. Wagner, "*Conceptual Multidimensional Data Model Based on Object-Oriented MetaCube",* Proceedings of the 2001 ACM Symposium on Applied Computing Las Vegas, Nevada, United States ,2001.

[12] Thanh Binh Nyguyen, A Min Tjoa, and Oscar Mangisengi, "*MetaCube-X: An XML MetaData Foundation for Interoperability Search Among Web Warehouses",* Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001). Switzerland, June 2001.

[13] N. N: "*XML for Analysis Specification. Version 1.0",* Microsoft Corporation, Hyperion Solutions Corporation, 2001.

[14] John Mikesell, *"Implementing the XML for Analysis Provider for SQL Server 2000 Analysis Services",* Microsoft Corporation. 2004.

[15] N. N: *"Designing and Implementing OLAP Solutions with Microsoft® SQL Server™ 2000, Course# 2074 Curriculum",* Microsoft Corporation, 2001

[16] N. N: *"Populating a Data Warehouse with Microsoft SQL Server™ 2000 Data Transformation Services",* Course# 2092 Curriculum, Microsoft Corporation, 2001

[17] Wolfgang Hümmer, Andreas Bauer, and Gunnar Harde., *"X- Cube-XML for Data Warehouses. Proceedings of the 6th ACM international Workshop on Data Warehousing and OLAP",* November 2003.

[18] R. A. Moeller. Distributed Data Warehousing Using Web Technology. AMACOM. 2001.

[19] N. N: Project-Team gemo: "*Management of Data and Knowledge Distributed Over the Web Activity Report".* INRIA, 2004

[20] Kimball Ralph, *"Laura Reeves, Margy Ross, and Warren Thornthwaite. The Data Warehouse Lifecycle Toolkit",* Second Edition John Wiley & Sons 2002.

[21] Kimball Ralph, and Margy Ross, *"The Data Warehouse Toolkit",* Second Edition John Wiley & Sons 2003.

[22] Kimball Ralph, and Richard Merz. *"The Data Webhouse Toolkit (Building the Web Enabled Data Warehouse)",* John Willey & Sons, 2000.

[23] Panos Vassiliadis, Timos Sellis. *"A Survey of Logical Models for OLAP Databases",* SIGMOD Record, Vol. 28, No. 4, December 1999.

[24] Micheleline Han, *"Data Mining Concepts and Techniques", Morgan Kaufmann",* 2004.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# PAFSV: A Process Algebraic Framework for SystemVerilog

K. L. Man
Centre for Efficiency-Oriented Languages (CEOL)
Department of Computer Science
University College Cork (UCC), Ireland
Email: pafsv.team@gmail.com

***Abstract*—We develop a process algebraic framework, called process algebraic framework for IEEE 1800$^{\text{TM}}$ SystemVerilog (PAFSV), for formal specification and analysis of IEEE 1800$^{\text{TM}}$ SystemVerilog designs. The formal semantics of PAFSV is defined by means of deduction rules that associate a time transition system with a PAFSV process. A set of properties of PAFSV is presented for a notion of bisimilarity. PAFSV may be regarded as the formal language of a significant subset of IEEE 1800$^{\text{TM}}$ SystemVerilog. To show that PAFSV is useful for the formal specification and analysis of IEEE 1800$^{\text{TM}}$ SystemVerilog designs, we illustrate the use of PAFSV with some examples: a MUX, a synchronous reset D flip-flop and an arbiter.**

## I. Introduction

THE goal of developing a formal semantics is to provide a complete and unambiguous specification of the language. It also contributes significantly to the sharing, portability and integration of various applications in simulation, synthesis and formal verification. *Formal languages* with a semantics formally defined in *Computer Science* increase understanding of systems and clarity of specifications; and help solving problems and remove errors. Over the years, several flavours of formal languages have been gaining industrial acceptance. *Process algebras* [1] are formal languages that have formal syntax and semantics for specifying and reasoning about different systems. They are also useful tools for verification of various systems. Generally speaking, process algebras describe the behaviour of processes and provide operations that allow to compose systems to obtain more complex systems. Moreover, the analysis and verification of systems described using process algebras can be partially or completely carried out by mathematical proofs using equational theory.

In addition, the strength of the field of process algebras lies in the ability to use *algebraic reasoning* [1] (also known as equational reasoning) that allows rewriting processes using axioms (e.g. for commutativity and associativity) to a simpler form. By using axioms, we can also perform calculations with processes. These can be advantageous for many forms of analysis. Process algebras have also helped to achieve a deeper understanding of the nature of concepts like observable behaviour in the presence of non-determinism, system composition by interconnection of system components modelled as processes in a parallel context and notions of behavioural equivalence (e.g. bisimulation [1]) of such systems.

Serious efforts have been made in the past to deal with systems (e.g. real-time systems [2] and hybrid systems [3], [4]) in a process algebraic way. Over the years, process algebras have been successfully used in a wide range of problems and in practical applications in both academia and industry for analysis of many different systems.

On the other hand, the need for a formal and well-defined semantics of a *Hardware Description Languages* (HDL) is widely accepted and desirable for architects, engineers and researchers in the electronic design community. IEEE 1800$^{\text{TM}}$ *SystemVerilog* [5] (in what follows, we abbreviate the IEEE 1800$^{\text{TM}}$ SystemVerilog as SystemVerilog) is the industry's first unified hardware description and verification language (HDVL) standard; and SystemVerilog is a major extension of the established IEEE 1364$^{\text{TM}}$ *Verilog* language [6] (see also [7]).

However, the standard semantics of SystemVerilog is informal. We believe that the fundamental tenets of process algebras are highly compatible with the behavioural approach of systems described in SystemVerilog. Hence, in this paper, we present a process algebraic framework called **PAFSV** that is suitable for modelling and analysis of systems described in SystemVerilog (in a formal way). **PAFSV** covers the main features of SystemVerilog (i.e. a significant subset of SystemVerilog) including decision statements and immediate assertions; and also aims to achieve a satisfactory level of abstraction and a more faithful modelling of concurrency. Although it is desirable and very important to have pure parallelism for hardware simulation, the SystemVerilog simulators "*in-use*" at this moment still implement parallelism via non-determinism.

Therefore, we realise that it is more fruitful to develop our process algebraic framework for SystemVerilog such that the execution of a system described in such a framework (**PAFSV**) consists of interleaving transitions from concurrent processes. Moreover, we adopt the view that a system described in **PAFSV** is a system in which an instantaneous state transition occurs on the system performing an action and a delay takes place on the system idling between performing successive actions. A technical advantage of our work is that, in contrast to other attempts to formalise semantics of SystemVerilog and HDLs, specifications described in **PAFSV** can be directly executable.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

The formal semantics of $\mathbf{PAFSV}$ is defined by means of deduction rules in a standard structured operational semantics (SOS) [9] style that associate a *Time Transition System* (TTS) with a $\mathbf{PAFSV}$ process. A set of properties of $\mathbf{PAFSV}$ is presented for a notion of bisimilarity. Overview of process algebras, Verilog and SystemVerilog is not given in this paper. Some familiarity with them is required. The desirable background can, for example, be found in [1], [5], [6].

Over the years, different formal approaches have been studied and investigated for VHDL [10], Verilog [11], [12] and SystemC [14], [15]. Most of these works could only be considered as theoretical frameworks, except a few trails ([13], [17]), because they are not executable. Research work in the formal semantics of SystemVerilog based on *Abstract State Machines* (ASMs) [16] and rewrite rules already exist [5]. Also, ASM specifications and rewrite rules are not directly executable. It is also generally believed that a structured operational semantics (SOS) provides more clear intuitions; and ASM specifications and rewrite rules appear to be less suited to describe the dynamic behaviour of processes.

Since processes are the basic units of execution within SystemVerilog that are used to simulate the behaviour of a system, a process algebraic framework in a SOS style is a more immediate choice to give the formal semantics of SystemVerilog (these motivated us to develop $\mathbf{PAFSV}$ in a process algebraic way with SOS deduction rules). Based on the similar motivations and needs, three years ago, $SystemC^{\mathbb{FL}}$ [17], [18], [19] (a timed process algebra) was introduced for formal specification and analysis of SystemC designs. $SystemC^{\mathbb{FL}}$ initiated an attempt to extend the knowledge and experience of the field of process algebras to SystemC designs. Clearly, SystemVerilog and SystemC are similar and our research work in this direction was highly inspired by the theoretical aspects of $SystemC^{\mathbb{FL}}$. Hence, a formal comparison between them is indispensable (as future work). Furthermore, an introduction (paper) of $\mathbf{PAFSV}$ can be found in [20]. Such a paper only informally presented the syntax and semantics of $\mathbf{PAFSV}$. Also, no deduction rules were given, validation of the semantics was not discussed and no analysis example of $\mathbf{PAFSV}$ specifications was provided.

This paper is organised as follows. Section II shows the goals, the data types, formal syntax and formal semantics of our process algebraic framework $\mathbf{PAFSV}$. To illustrate the use, effectiveness and applicability of the deduction rules, in Section III, some simple specifications of $\mathbf{PAFSV}$ are provided. In Section IV, the correctness of the formal semantics of $\mathbf{PAFSV}$ defined in Subsection II-E is discussed; and a notion of equivalence is defined, which is shown to be a congruence for all $\mathbf{PAFSV}$ operators. Also, a set of useful properties of closed $\mathbf{PAFSV}$ process terms is given in the same section. Samples (modelling some SystemVerilog designs) of the application of $\mathbf{PAFSV}$ are shown in Section V. A formal analysis (by means of a complete mathematical proof) of a SystemVerilog design via $\mathbf{PAFSV}$ is presented in Section VI. Finally, concluding remarks are made in Section VII and the direction of future work is pointed out in the same section.

## II. $\mathbf{PAFSV}$

Obviously, it is not possible to cover all the aspects of SystemVerilog and define a process algebraic framework for it in one paper. Hence, in this section, we outline the goals to achieve in this paper.

We propose a process algebraic framework namely $\mathbf{PAFSV}$ that has a formal and compositional semantics based on a time transition system for formal specification and analysis of SystemVerilog designs. The intention of our process algebraic framework $\mathbf{PAFSV}$ is as follows:

- to give a formal semantics to a significant subset of SystemVerilog using the operational approach of [9];
- to serve as a mathematical basis for improvement of design strategies of SystemVerilog and possibilities to analyse SystemVerilog designs;
- to serve as a coherent first step for a semantics interoperability analysis on semantics domain such as SystemC and $SystemC^{\mathbb{FL}}$;
- to initiate an attempt to extend the knowledge and experience of the field of process algebras to SystemVerilog designs;
- to be used as the formal language for a significant subset of SystemVerilog.

### A. Data types

In order to define the semantics of processes, we need to make some assumptions about the data types:

1) Let $\mathrm{Var}$ denote the set of all variables $(x_0, \ldots, x_n, \mathtt{time})$. Besides the variables $x_0, \ldots, x_n$, the existence of the predefined reserved global variable `time` which denotes the current time, the value of which is initially zero, is assumed. This variable cannot be declared.
2) Let $\mathrm{Value}$ denote the set of all possible values $(v_0, \ldots, v_m, \perp)$ that contains at least all $\mathrm{Integers}$, all $\mathrm{Reals}$, all $\mathrm{Shortreals}$, all $2-\mathrm{statevalues}$ and all $4-\mathrm{statevalues}$ as defined in SystemVerilog (see [5] for details); all $\mathrm{Booleans}$ and $\perp$, where $\perp$ denotes the "*undefinedness*".
3) We then define a *valuation* as a partial function from variables to values. Syntactically, a valuation is denoted by a set of pairs $\{x_0 \mapsto v_0, \ldots, x_n, \mapsto v_n, \mathtt{time} \mapsto t\}$, where $x_i$ represents a variable and $v_i$ its associating value; and $t \in \mathbb{R}_{\geq 0}$.
4) Further to this, the set of all valuations is denoted by $\Sigma$.

Note that the type "$\mathrm{array}$" in SystemVerilog is not formalised yet in $\mathbf{PAFSV}$. However, the behaviour of elements in an array in SystemVerilog can be modelled in $\mathbf{PAFSV}$ by introducing fresh variables. As an example, for an array A[0:10] in SystemVerilog, we can introduce fresh variables $A_0, \ldots, A_{10}$ in $\mathbf{PAFSV}$ to associate correspondingly A[0] with $A_0$, A[1] with $A_1$ and so on.

### B. Formal syntax

To avoid confusion with the definition of a process in SystemVerilog, it is important to clearly state that, in our process algebraic framework **PAFSV**, we choose the terminology "***a process term***" as a formal term (generated restrictively through the formal syntax of **PAFSV**) to describe the possible behaviour of a **PAFSV** process (see Subsection II-E) and not a process as defined in SystemVerilog.

Furthermore, process terms $p \in P$ are the core elements of the **PAFSV**. The semantics of those process terms is defined in terms of the core process terms given in this subsection. The set of process terms $P$ is defined according to the following grammar for the process terms $p \in P$:

$$\begin{array}{rll} p \ ::= & \textbf{deadlock} \ \mid \ \textbf{skip} \ \mid \ x := e \\ \mid & \textbf{delay}(n) \ \mid \ \textbf{any}\ p \ \mid \ \textbf{if}(b)\ p\ \textbf{else}\ p \\ \mid & p;\ p \ \mid \ \textbf{wait}(b)\ p \ \mid \ \textbf{while}(b)\ p \\ \mid & \textbf{assign}\ w := e \ \mid \ @_{(\eta_1(l_1),\ldots,\eta_n(l_n))}\ p \\ \mid & p \circledast p \ \mid \ p \parallel p \ \mid \ \textbf{repeat}\ p \\ \mid & \textbf{assert}(b)\ p \ \mid \ p\ \textbf{disrupt}\ p \end{array}$$

Here, $x$ and $w$ are variables taken from $\mathrm{Var}$ and $n \in \mathbb{R}_{\geq 0}$. $b$ and $e$ denote a boolean expression and an expression over variables from $\mathrm{Var}$, respectively. Moreover, $\eta_1, \ldots, \eta_n$ represent boolean functions with corresponding parameters $l_1, \ldots, l_n \in \mathrm{Var}$.

In **PAFSV**, we allow the use of common arithmetic operators (e.g. $+, -$), relational operators (e.g. $=, \geq$) and logical operators (e.g. $\wedge, \vee$) as in mathematics to construct expressions over variables from $\mathrm{Var}$. The operators are listed in descending order of their binding strength as follows: $\{$**if**(_)**_else_**, **wait**(_)_, **while**(_)_, **assert**(_)_$\}$, _; _, **_disrupt_**, $\{\_ \circledast \_, \_ \parallel \_\}$. The operators inside the braces have equal binding strength. In addition, operators of equal binding strength associate to the right, and parentheses may be used to group expressions. For example, $p;\ q;\ r$ means $p;\ (q;\ r)$, where $p, q, r \in P$. Apart from process terms: **deadlock**, **skip**, **any_**, **_disrupt_**, and $\_ \circledast \_$, all other syntax elements in **PAFSV** are the formalisation of the corresponding language elements (based on classical process algebra tenets) in SystemVerilog.

Process terms **deadlock** and **skip**; and operator $\_ \circledast \_$ are mainly introduced for calculation and axiomatisation purposes. The **any_** operator was originally introduced in Hybrid Chi [3] (to be precise, in Hybrid Chi, such an operator is called "*the any delay operator*" and denoted by "[ ]"). It is used to give an arbitrary delay behaviour to a process term. We can make use of this operator to simplify our deduction rules in a remarkable way. The **_disrupt_** is inspired by the analogy of the disrupt operator in HyPA [4]. This can be used to model event controls in **PAFSV** in a very efficient way. A concise explanation of the formal syntax of **PAFSV** is given below. Subsection II-E gives a more detailed account of its meaning.

### C. Atomic process terms

The atomic process terms of **PAFSV** are process term constructors that cannot be split into smaller process terms. They are:

1) The *deadlock* process term **deadlock** is introduced as a constant, which represents no behaviour. This means that it cannot perform any actions or delays.
2) The *skip* process term **skip** can only perform the internal action $\tau$ to termination, which is not externally visible.
3) The *procedural assignment* process term $x := e$ assigns the value of expression $e$ to variable $x$ (in an atomic way).
4) The *continuous assignment* process term **assign** $w := e$ continuously watches for changes of the variables that occur on the expression $e$. Whenever there is a change, the value of $e$ is re-evaluated and then propagated it immediately to $w$.
5) The *delay* process term **delay**$(n)$ denotes a process term that first delays for $n$ time units, and then terminates by means of the internal action $\tau$.

### D. Operators

Atomic process terms can be combined using the following operators. The operators are:

1) By means of the application of the *any* operator to process term $p \in P$ (i.e. **any** $p$), delay behaviour of arbitrary duration can be specified. The resulting behaviour is such that arbitrary delays are allowed. As a consequence, any delay behaviour of $p$ is neglected. The action behaviour of $p$ remains unchanged. This operator can even be used to add arbitrary behaviour to an undelayable process term.
2) The *if_else* process term **if**$(b)$ $p$ **else** $q$ first evaluates the boolean expression $b$. If $b$ evaluates to *true*, then $p$ is executed, otherwise $q \in P$ is executed.
3) The *sequential composition* of process terms $p$ and $q$ (i.e. $p;\ q$) behaves as process term $p$ until $p$ terminates, and then continues to behave as process term $q$.
4) The *wait* process term **wait**$(b)$ $p$ can perform whatever $p$ can perform under the condition that the boolean expression $b$ evaluates to $true$. Otherwise, it is blocked until $b$ becomes $true$.
5) Similarly, the *while* process term **while**$(b)$ $p$ can perform whatever $p$ can do under the condition that the boolean expression $b$ evaluates to $true$ and then followed by the original iteration process term (i.e. **while**$(b)$ $p$). In case $b$ evaluates to $false$, the while process term **while**$(b)$ $p$ terminates by means of the internal action $\tau$.
6) The *event* process term $@_{(\eta_1(l_1),\ldots,\eta_n(l_n))}\ p$ can perform whatever $p$ can do under the condition that any of the boolean functions $\eta_1(l_1), \ldots, \eta_n(l_n)$ returns to $true$. If there is no such a function, $p$ will be triggered by $\eta_1(l_1), \ldots, \eta_n(l_n)$. Intuitively, functions $\eta_1, \ldots, \eta_n$ are used to model event changes as event controls $levelchange$, $posedge$ and $negedge$ in SystemVerilog.
7) The *alternative composition* of process terms $p$ and $q$ (i.e. $p \circledast q$) allows a non-deterministic choice between different actions of the process term either $p$ or $q$.
8) The *parallel composition* of process terms $p$ and $q$ (i.e. $p \parallel q$) executes $p$ and $q$ concurrently in an interleaved fashion. For the time behaviour, the participants in the parallel composition have to synchronise.

9) The *repeat* process term **repeat** $p$ represents the infinite repetition of process term $p$. Note that the idea behind the *repeat* statement in SystemVerilog is slightly different from **repeat** $p$ in $\mathbf{PAFSV}$. The repeat statement specifies the number of times of a loop to be repeated. The same goal can be achieved by using the repeat process term in combination with the if_else process term in $\mathbf{PAFSV}$.
10) The *assert* process term **assert**$(b)$ $p$ checks immediately the property $b$ (expressed as a boolean expression). If $b$ holds, $p$ is executed.
11) The *disrupt* process term $p$ **disrupt** $q$ intends to give priority of the execution of process term $p$ over process term $q$. The need and use of this operator will be illustrated in Section VI.

### E. Formal semantics

In this subsection, we give a formal semantics to the syntax defined for $\mathbf{PAFSV}$ in the previous subsection, by constructing a kind of time transition system (TTS), for each process term and each possible valuation of variables.

**Definition 1** *We use the convention $\langle p, \sigma\rangle$ to write a $\mathbf{PAFSV}$ process, where $p \in P$ and $\sigma \in \Sigma$.*

**Definition 2** *The set of actions $A_\tau$ contains at least $aa(x, v)$ and $\tau$, where $aa(x, v)$ is the assignment action (i.e. the value of $v$ is assigned to $x$) and $\tau$ is the internal action. The set $A_\tau$ is considered as a parameter of $\mathbf{PAFSV}$ that can be freely instantiated.*

**Definition 3** *We give a formal semantics for $\mathbf{PAFSV}$ processes in terms of a time transition system (TTS), and define the following transition relations on processes of $\mathbf{PAFSV}$:*

- $\_ \xrightarrow{\_} \langle \checkmark, \_\rangle \subseteq (P \times \Sigma) \times A_\tau \times \Sigma$, *denotes termination, where $\checkmark$ is used to indicate a successful termination, and $\checkmark$ is not a process term;*
- $\_ \xrightarrow{\_} \_ \subseteq (P \times \Sigma) \times A_\tau \times (P \times \Sigma)$, *denotes action transition;*
- $\_ \overset{\_}{\longmapsto} \_ \subseteq (P \times \Sigma) \times \mathbb{R}_{>0} \times (P \times \Sigma)$, *denotes time transition (so-called delay).*

For $p, p' \in P$; $\sigma, \sigma' \in \Sigma$, $a \in A_\tau$ and $d \in \mathbb{R}_{>0}$, the three kinds of transition relations can be explained as follows:

1) Firstly, a termination $\langle p, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle$ is that the process executes the action $a$ followed by termination.
2) Secondly, an action transition $\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle$ is that the process $\langle p, \sigma\rangle$ executes the action $a$ starting with the current valuation $\sigma$ and by this execution $p$ evolves into $p'$, where $\sigma'$ represents the accompanying valuation of the process after the action $a$ is executed.
3) Thirdly, a time transition $\langle p, \sigma\rangle \overset{d}{\longmapsto} \langle p', \sigma'\rangle$ is that the process $\langle p, \sigma\rangle$ may idle during a $d$ time units and then behaves like $\langle p', \sigma'\rangle$.

### F. Deduction rules

The above transition relations are defined through deduction rules (SOS style). These rules (of the form $\frac{\textit{permises}}{\textit{conclusions}}$) have two parts: on the top of the bar we put *premises* of the rule, and below it the *conclusions*. If the premise(s) hold(s), then we infer that the conclusion(s) hold(s) as well. In case there is no premise, the deduction rule becomes an axiom.

Apart from the syntax restriction as already shown in Subsection II-B (e.g. $x, w \in \mathrm{Var}$), for all deduction rules, we further require that $p, q, p', q' \in P$; $\sigma, \sigma', \sigma'' \in \Sigma$; $a, b \in A_\tau$, $d \in \mathbb{R}_{>0}$, $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma') = \mathrm{dom}(\sigma'')$; $\sigma$, $\sigma'$, $\sigma''$ and $\bar{\sigma}(e)$ are defined, where the notation $\bar{\sigma}(e)$ is used to represent the value of expression $e$ in $\sigma$.

Also, we make use of the sets of variables $\mathrm{Var}^- = \{x^- \mid x \in \mathrm{Var}\}$ and $\mathrm{Var}^+ = \{x^+ \mid x \in \mathrm{Var}\}$, modelling the current and future value of a variable, respectively. Similarly, $e^-$ and $e^+$ are used to represent the current and future value of $e$ respectively.

In order to increase the readability of the $\mathbf{PAFSV}$ deduction rules, the notation $\overset{z}{\twoheadrightarrow}$ is used as a short-hand for $\xrightarrow{a}$ and $\overset{d}{\longmapsto}$.

***Deduction rules:*** It is not our intention to define deduction rules for all inductive cases for all operators in this paper. For simplicity, only relevant deduction rules for the use of this paper are shown in this subsection.

*1) Procedural assignment:*

$$\frac{}{\langle x := e, \sigma\rangle \xrightarrow{aa(x, \bar{\sigma}(e))} \langle \checkmark, \sigma[\bar{\sigma}(e)/x]\rangle}\ 1$$

By means of a procedural assignment (see Rule 1), the value of $e$ is assigned to $x$. Notice that $\sigma[\bar{\sigma}(e)/x]$ denotes the update of valuation $\sigma$ such that the new value of variable $x$ is $\bar{\sigma}(e)$.

*2) Sequential composition:*

$$\frac{\langle p, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle}{\langle p;\ q, \sigma\rangle \xrightarrow{a} \langle q, \sigma'\rangle}\ 2 \qquad \frac{\langle p, \sigma\rangle \overset{z}{\twoheadrightarrow} \langle p', \sigma'\rangle}{\langle p;\ q, \sigma\rangle \overset{z}{\twoheadrightarrow} \langle p';\ q, \sigma'\rangle}\ 3$$

The process term $q$ is executed after (successful) termination of the process term $p$ as defined by Rules 2 and 3.

*3) Parallel composition:*

$$\frac{\langle p, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle}{\langle p \parallel q, \sigma\rangle \xrightarrow{a} \langle q, \sigma'\rangle}\ 4 \qquad \frac{\langle q, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle}{\langle p \parallel q, \sigma\rangle \xrightarrow{a} \langle p, \sigma'\rangle}\ 5$$

$$\frac{\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle}{\langle p \parallel q, \sigma\rangle \xrightarrow{a} \langle p' \parallel q, \sigma'\rangle}\ 6 \qquad \frac{\langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle}{\langle p \parallel q, \sigma\rangle \xrightarrow{a} \langle p \parallel q', \sigma'\rangle}\ 7$$

$$\frac{\langle p, \sigma\rangle \overset{d}{\longmapsto} \langle p', \sigma'\rangle,\ \langle q, \sigma\rangle \overset{d}{\longmapsto} \langle q', \sigma'\rangle}{\langle p \parallel q, \sigma\rangle \overset{d}{\longmapsto} \langle p' \parallel q', \sigma'\rangle}\ 8$$

The parallel composition of the process terms $p$ and $q$ (i.e. $p \parallel q$) has as its behaviour with respect to action transitions the interleaving of the behaviours of process terms $p$ and $q$ (see from Rule 4 to Rule 7). If both process terms $p$ and $q$ can perform the same delay, then the parallel composition of process terms $p$ and $q$ (i.e. $p \parallel q$) can also perform that delay, as defined by Rule 8.

## III. Examples

Deduction rules offer preciseness, because they come with a mathematically defined semantics. Formal specifications can be analysed using deduction rules providing an absolute notion of correctness.

Also, these deduction rules can ensure the correctness of **PAFSV** specifications and can help modellers to make correct specifications.

In order to demonstrate the effectiveness and applicability of the deduction rules, two toy specifications in **PAFSV** are given in this section. These specifications also show how (illustrated by means of transition traces) process evolves during transitions.

Using the deduction rules, for instance, we can show that:

1) the process $\langle x := 5;\ y := 7, \{x \mapsto 0, y \mapsto 1\}\rangle$ can terminate successfully after a finite number of transitions.
   - ***Transition traces:*** According to Rule 1, the process $\langle x := 5, \{x \mapsto 0, y \mapsto 1\}\rangle$ can always perform an assignment action to a terminated process as follows: $\langle x := 5, \{x \mapsto 0, y \mapsto 1\}\rangle \xrightarrow{aa(x,5)} \langle \checkmark, \{x \mapsto 5, y \mapsto 1\}\rangle$. Due to this, we can apply Rule 3 to obtain $\langle x := 5;\ y := 7, \{x \mapsto 0, y \mapsto 1\}\rangle \xrightarrow{aa(x,5)} \langle y := 7, \{x \mapsto 5, y \mapsto 1\}\rangle$. Applying Rule 1 again, we have $\langle y := 7, \{x \mapsto 5, y \mapsto 1\}\rangle \xrightarrow{aa(y,7)} \langle \checkmark, \{x \mapsto 5, y \mapsto 7\}\rangle$.
2) the process $\langle (x := 1 \parallel y := 2);\ z := 3, \sigma\rangle$ cannot terminate successfully in two transitions.
   - ***Semantical proof:*** We assume to have $\langle (x := 1 \parallel y := 2);\ z := 3, \sigma\rangle \xrightarrow{a} \langle z := 3, \sigma'\rangle$ for some $a$ and $\sigma'$ in such a way that the process can terminate successfully in two transitions. This means that we must have the action transition $\langle x := 1 \parallel y := 2, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle$ as a premise necessarily for Rule 2. However, this is not possible due to Rules 4 and 5.

## IV. Validation of the Semantics

This section first shows that the term deduction system of **PAFSV** is well-defined. Then a notion of equivalence called *Stateless Bisimilarity* is defined (see also [3], [21]).

It is also shown that this relation is an equivalence and a *Congruence* [1] (which also means that compositionality preserved operationally in **PAFSV**) for all **PAFSV** operators.

A set of useful properties of **PAFSV** is sound with respect to the stateless bisimilarity that is also introduced.

### A. Well-definedness of the semantics

The deduction rules defined for **PAFSV** constitute a *Transition System Specification* (TSS) as described in [22]. The transitions that can be proven from a TSS define a time transition system (TTS).

The TTS of **PAFSV** contains terminations, action transitions and time transitions that can be proven from the deduction rules. In general, TSSs with negative premises[1] might not be *meaningful* (see [22] for details).

As we know that no negative premise is used in our deduction rules for **PAFSV**. So, it is not hard to see that the term deduction system of **PAFSV** is well-defined. This means that the system defines a unique transition system for each closed process term of **PAFSV**.

### B. Bisimilarity

Two closed **PAFSV** process terms are considered equivalent if they have the same behaviour (in the bisimulation sense) from the current state.

We also assume that the valuation (of the current state) contains at least the free occurrences of variables in the two closed **PAFSV** process terms being equivalent.

**Definition 4 (Stateless bisimilarity)** *A stateless bisimilarity on closed process terms is a relation $R \subseteq P \times P$ such that $\forall (p,q) \in R$, the following holds:*

1) $\forall \sigma, a, \sigma' : \langle p, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle \Leftrightarrow \langle q, \sigma\rangle \xrightarrow{a} \langle \checkmark, \sigma'\rangle$,
2) $\forall \sigma, a, p', \sigma' : \langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle \Rightarrow \exists q' : \langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle \wedge (p', q') \in R$,
3) $\forall \sigma, a, q', \sigma' : \langle q, \sigma\rangle \xrightarrow{a} \langle q', \sigma'\rangle \Rightarrow \exists p' : \langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle \wedge (p', q') \in R$,
4) $\forall \sigma, d, p', \sigma' : \langle p, \sigma\rangle \overset{d}{\longmapsto} \langle p', \sigma'\rangle \Rightarrow \exists q' : \langle q, \sigma\rangle \overset{d}{\longmapsto} \langle q', \sigma'\rangle \wedge (p', q') \in R$,
5) $\forall \sigma, d, q', \sigma' : \langle q, \sigma\rangle \overset{d}{\longmapsto} \langle q', \sigma'\rangle \Rightarrow \exists p' : \langle p, \sigma\rangle \overset{d}{\longmapsto} \langle p', \sigma'\rangle \wedge (p', q') \in R$.

*Two closed process terms $p$ and $q$ are stateless bisimilar, denoted by $p \underline{\leftrightarrow} q$, if there exists a stateless bisimilarity relation $R$ such that $(p,q) \in R$.*

Stateless bisimilarity is proved to be a congruence with respect to all **PAFSV** operators. As a consequence, algebraic reasoning is facilitated, since it is allowed to replace equals by equals in any context.

**Theorem 1 (Congruence)** *Stateless bisimilarity is a congruence with respect to all* **PAFSV** *operators.*

***Proof:*** *All deduction rules of* **PAFSV** *are in the process-tyft format of [21]. It follows from [21] that stateless bisimilarity is a congruence.*

### C. Properties

In this subsection, some properties of the operators of **PAFSV** that hold with respect to stateless bisimilarity are discussed. Most of these correspond well with our intuitions, and hence this can be considered as an additional validation of the semantics.

It is not our intention to provide a complete list of such properties (complete in the sense that every equivalence between closed process terms is derivable from those properties).

[1] We write a negative premise for action transition as $\langle p, \sigma\rangle \overset{a}{\nrightarrow}$ for the set of all transitions formulas $\neg(\langle p, \sigma\rangle \xrightarrow{a} \langle p', \sigma'\rangle)$, where $p, p' \in P$, $a \in A_\tau$ and $\sigma, \sigma' \in \Sigma$. In a similar way, we can define negative premises for termination and time transition.

**Proposition 1 (Properties)** *A set of properties is introduced for **PAFSV** described in this paper for $p, q, r \in P$. These properties are sound with respect to the stateless bisimilarity.*

1) ***skip*** $\underline{\leftrightarrow}$ ***delay***$(0)$,
2) ***deadlock***$;\ p \ \underline{\leftrightarrow}$ ***deadlock****,*
3) $(p;\ q);\ r \ \underline{\leftrightarrow}\ p;\ (q;\ r)$,
4) ***any*** $p;\ q \ \underline{\leftrightarrow}$ ***any*** $(p;\ q)$,
5) $p \circledast q \ \underline{\leftrightarrow}\ q \circledast p$,
6) $(p \circledast q);\ r \ \underline{\leftrightarrow}\ p;\ r \circledast q;\ r$,
7) $(p \circledast q) \circledast r \ \underline{\leftrightarrow}\ p \circledast (q \circledast r)$,
8) $p \parallel q \ \underline{\leftrightarrow}\ q \parallel p$,
9) $(p \parallel q) \parallel r \ \underline{\leftrightarrow}\ p \parallel (q \parallel r)$,
10) ***any*** $p \circledast$ ***any*** $q \ \underline{\leftrightarrow}$ ***any*** $(p \circledast q)$,

***Proof:*** *We leave out the proofs, because most of the proofs are proofs for distributivity, commutativity and associativity as in classical process algebras. Similar proofs can also be found in [3].*

### Intuition behind the properties

The intuition of the above properties is as follows:

- Since **skip** and **delay**$(0)$ can only perform the internal action $\tau$ to termination, both process terms are equivalent.
- A deadlock process term followed by some other process terms is equivalent to the **deadlock** itself because the deadlock process term does not terminate successfully, i.e. **deadlock** is a left-zero element for sequential composition.
- Sequential composition is associative.
- The any operator distributes to the right argument of a sequential composition.
- Alternative composition and parallel composition are commutative and associative.
- Alternative composition distributes over sequential composition from the left, but not from the right.
- The any operator distributes over the alternative composition.

## V. Examples of **PAFSV** Specifications

This section is a sample of the application of **PAFSV**. It is meant to give a first impression of how one can describe the behaviour of some SystemVerilog designs in **PAFSV** (in a complete mathematical sense). We describe the behaviour of a simple MUX and a simple synchronous reset D flip-flop.

### *A. MUX*

In electronic designs, a multiplexer (MUX) is a device that encodes information from two or more data inputs into a single output (i.e. multiplexers function as multiple-inputs and single-output switches). A multiplexer described below (in SystemVerilog) has two inputs and a selector that connects a specific input to the single output. Figure 2 depicts such a MUX.

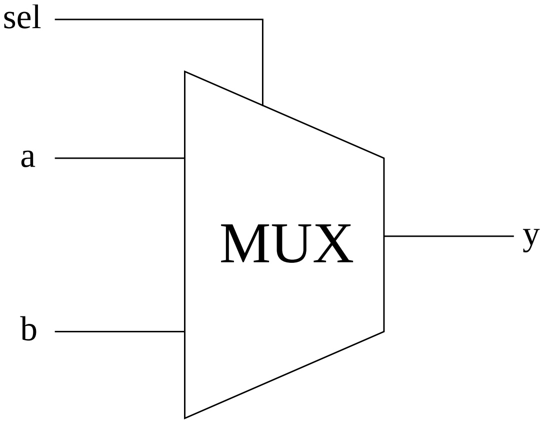


Fig. 1. A MUX.

```
module  simple_mux (
input  wire  a,
input  wire  b,
input  wire  sel,
output wire  y
);
assign y = (sel) ? a : b;
endmodule
```

The formal **PAFSV** specification (as a process term) below can be regarded as the (formal) mathematical expression of the above multiplexer (described as a SystemVerilog module):

$$\textbf{if}(sel)\ y := a\ \textbf{else}\ y := b$$

Needless to mention that, in SystemVerilog, the conditional operator “(condition) ? (result if true):(result if false)” can be considered as an **if(_)_else_** statement. In the **PAFSV** specification, an if_else process term is used to model the behaviour of such a MUX.

### *B. Synchronous reset D flip-flop*

Synchronous reset D flip-flops are among the basic building blocks of RTL designs. A synchronous reset D flip-flop has a clock input ($clk$) in the event list, a data input ($d$), a reset ($rst$) and a data output ($Q$). Figure 2 depicts such a synchronous reset D flip-flop.

A synchronous reset D flip-flop described below (as a module in SystemVerilog) is inferred by using posedge clause for the clock $clk$ in the event list.

```
module dff_sync_reset (
input  wire d,
input  wire clk,
input  wire rst,
output reg  Q
);
always_ff @ (posedge clk)
if (~reset) begin
  Q = 1’b0;
end  else begin
  Q = d;
end
endmodule
```

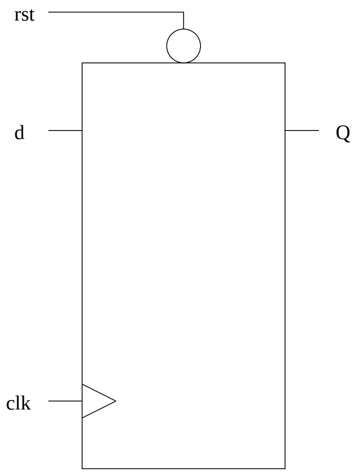


Fig. 2. A synchronous reset D flip-flop.

The formal **PAFSV** specification (as a process term) of the above synchronous reset D flip-flop (described as a module in SystemVerilog) is given as follows:

$$\begin{array}{rl} \mathrm{DFF} & \approx \textbf{repeat}(@_{(\eta_{negedge}(clk))}\mathrm{OUT}) \\ \mathrm{OUT} & \approx \textbf{if}(\neg rst)\ Q := 1'b0\ \textbf{else}\ Q := d \end{array}$$

In the **PAFSV** specification (i.e. process term DFF), the behaviour of the synchronous reset D flip-flop is modelled by means of the if_else process term using “$\neg rst$ (active low reset)” as the condition of such a process term.

This if_else process term is further triggered repeatedly by the event process term, which is positively sensitive to the clock (i.e. $clk$).

## VI. Analysis of an PAFSV Specification

We have already shown in Section V that **PAFSV** specifications can be used to formally represent SystemVerilog designs. Therefore, in this section, we formally analyse a simple arbiter described in SystemVerilog via **PAFSV**.

### A. An arbiter

Arbiter circuits are standard digital hardware verification benchmark circuits. In general, the role of an arbiter is to grant access to the shared resource by raising the corresponding $grant$ signal and keeping it that way until the $request$ signal is removed.

A test for the arbiter can be generated by an immediate assertion as follows:

$$\text{“}assertion : grant \wedge request\text{”}.$$

This immediate assertion can be considered as a *liveness property* of the arbiter. If the assertion holds, this means that the arbiter works as expected. Below is a SystemVerilog design of the simple arbiter as described above:

```
module assert_immediate();
reg clk, grant, request;
time current_time;
initial begin
   clk = 0;
   grant = 0;
   request = 0;
   #4 request = 1;
   #4 grant = 1;
   #4 request = 0;
   #4 $finish;
end
always #5 clk = ~ clk;
always @ (negedge clk)
begin
if (grant == 1) begin
 CHECK_REQ_WHEN_GNT:
   assert(grant && request) begin
   current_time = $time;
    $display {‘‘Works as expected’’);
    end
 end
endmodule
```

A formal **PAFSV** specification of the above SystemVerilog arbiter is given as follows:

$$\langle\ \mathrm{INIT} \parallel \mathrm{ARB} \parallel \mathrm{CLK}\ \textbf{disrupt}\ \mathrm{ASSER}, \sigma\ \rangle, \text{ where}$$

$$\begin{array}{cl} \mathrm{INIT} & \approx clk := 0;\ grant := 0;\ request := 0 \\ \mathrm{ARB} & \approx \mathrm{R}_1;\ \mathrm{G};\ \mathrm{R}_0;\ \mathrm{S} \\ \mathrm{R}_1 & \approx \textbf{delay}(4);\ request := 1 \\ \mathrm{G} & \approx \textbf{delay}(4);\ grant := 1 \\ \mathrm{R}_0 & \approx \textbf{delay}(4);\ request := 0 \\ \mathrm{S} & \approx \textbf{delay}(4);\ \textbf{skip} \\ \mathrm{CLK} & \approx \textbf{repeat}(\textbf{delay}(5);\ clk := \neg clk) \\ \mathrm{ASSER} & \approx \textbf{repeat}(@_{(\eta_{negedge}(clk))}\mathrm{PROP};\ \textbf{skip}) \\ \mathrm{PROP} & \approx \textbf{assert}(grant \wedge request)\ t := \mathtt{time} \end{array}$$

$$\sigma = \{clk \mapsto \bot, grant \mapsto \bot, request \mapsto \bot, t \mapsto \bot, \mathtt{time} \mapsto 0\}.$$

The formal specification of the arbiter is a parallel composition of process terms INIT, ARB and CLK **disrupt** ASSER:

- INIT - It assigns the initial values to variables $clk$, $grant$ and $request$ (i.e. the initialisation).
- ARB - It models the change of behaviour of variables $clk$, $grant$ and $request$ according to time.
- CLK - It models the behaviour of a clock (i.e. $clk$) which swaps the values between “0” and “1” every 5 time units.
- ASSER - It expresses the immediate assertion for the arbiter (as indicated above).
- CLK **disrupt** ASSER - It models the fact that the test of the immediate assertion is executed whenever there is a negative change in $clk$. When this happens, the current time is assigned to the variable $t$. ***Remark:*** This also explains the need and the use of the “**disrupt** process term”, because the execution of process term CLK must have a higher priority than the execution of process term ASSER (since the change of the clock causes the test to be run).

### B. Formal analysis of the arbiter

The arbiter described in **PAFSV** was analysed by means of a complete mathematical proof via transition traces according to deduction rules of **PAFSV**. The liveness property (i.e. the immediate assertion holds at least for some times) of the arbiter was proved to hold. In this paper, due to the reason of spaces, the above-mentioned proof is omitted.

## VII. Conclusions and Future Work

In order to illustrate our work clearly, only simple examples were given in this paper. Nevertheless, the use of **PAFSV** is generally applicable to all sizes and levels of SystemVerilog designs. Nevertheless, we reached our goals (as indicated in Section II). We also believe that our process algebraic framework **PAFSV** can serve as a mathematical basis for improvement of the design strategies of SystemVerilog, and possibilities to analyse SystemVerilog designs, because **PAFSV**

1) comprises mathematical expressions for SystemVerilog;
2) allows for analysis of specifications in a compositional way;
3) allows for equational reasoning on specifications;
4) contributes significantly to the investigation of interoperabilites of SystemVerilog with SystemC and $SystemC^{\mathbb{FL}}$.

We have the idea that, like $SystemC^{\mathbb{FL}}$, **PAFSV** can serve as a *single-formalism-multi-solution*. This means that we can formally translate a **PAFSV** specification to the input languages (e.g. SMV [23], Promela [24] and timed automata [25]) of several verification tools (e.g. SMV [23], SPIN [24] and Uppaal [26]) and it can be verified in those verification tool environments.

Our future work will develop/investigate such translations. For practical applications, we will apply **PAFSV** to formally represent SystemVerilog designs (for formal analysis purposes) in the design flow of the project: "$\mathcal{MOQA}$ Processor: An Entirely New Type of Processor for Modular Quantitative Analysis" as reported in [27].

## VIII. Availability

The full set of **PAFSV** deduction rules and the complete mathematical proof of the correctness of the arbiter (see VI-B for details) are available by email at pafsv.team@gmail.com.

## IX. Acknowledgement

The author wishes to thank Jos Baeten, Bert van Beek, Mohammad Mousavi, Koos Rooda, Ramon Schiffelers, Pieter Cuijpers, Michel Reniers, Kees Middelburg, Uzma Khadim and Muck van Weerdenburg for many stimulating and helpful discussions focusing on process algebras for distinct systems in the past few years.

Many thanks go to Michel Schellekens and Menouer Boubekeur for their contributions to the introduction paper of **PAFSV** (see [20]) and the industrial collaborators Solari—Hong Kong (http://www.solari-hk.com/), International Software and Productivity Engineering Institute—USA (http://www.intspei.com), Intelligent Support Ltd.—United Kingdom (http://www.isupport-ltd.co.uk) and Minteos—Italy (http://www.minteos.com) of the research work presented in this paper.

## References

[1] J. C. M. Baeten, W. P. Weijland, *Process Algebra*, Number 18 in Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1990.

[2] J. C. M. Baeten, C. A. Middelburg, *Process Algebra with Timing*, in EATCS Monographs Series, Springer-Verlag, 2002.

[3] D. A. van Beek, K. L. Man, M. A. Reniers, J. E. Rooda, R. R. H. Schiffelers, *Syntax and Consistent Equation Semantics of Hybrid Chi*, in Journal of Logic and Algebraic Programming, 68(1–2):129-210, 2006.

[4] P. J. L. Cuijpers, M. A. Reniers, *Hybrid Process Algebra*, in Journal of Logic and Algebraic Programming, 62(2):191–245, 2005.

[5] *IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language*, IEEE Std $1800^{\mathrm{TM}}$-2005, IEEE Computer Society, 2005.

[6] *IEEE Standard for Verilog Hardware Description Language*, IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001), IEEE Computer Society, 2006.

[7] *SystemVerilog 3.1a: Accellera's Extensions to Verilog*, Napa, CA, 2003. Available in PDF form at http://www.systemverilog.com/

[8] **PAFSV** *homepage*, http://digilander.libero.it/systemcfl/pafsv/

[9] G. D. Plotkin, *A Structural Approach to Operational Semantics*, in Report DAIMI FN-0.59, Computer Science Department, Aarhus University, 1981.

[10] P. T. Breuer, C. Delgado Kloos, *Formal Semantics for VHDL*, Kluwer Academic Publishers, 1995.

[11] G. Schneider, X. Qiwen, *Towards an Operational Semantics of Verilog*, UNU/IIST Report No. 147, International Institute for Software Technology, United Nations University, Macau, 1998.

[12] G. Schneider, X. Qiwen, *Towards a Formal Semantics of Verilog Using Duration Calculus*, IN A. Ravn, H. Rischel, editors, Formal Techniques for Real-Time and Fault Tolerant Systems (FTRTFT'98), LNCS, Springer-Verlag, 1998.

[13] J. Bowen, *Animating the Semantics of Verilog Using Prolog*, UNU/IIST Report No. 176, International Institute for Software Technology, United Nations University, Macau, 1999.

[14] W. Mueller, J. Ruf, D. Hofmann, J. Gerlach, T. Kropf and W.Rosenstiehl, *The Simulation Semantics of SystemC*, in Proceedings of DATE, 2001

[15] A. Salem, *Formal Semantics of Synchronous SystemC*, in Proceedings of DATE, 2003.

[16] W. Mueller, M. Zambaldi, W. Ecker, T. Kruse, *The Formal Simulation Semantics of SystemVerilog*, in Proceedings of the FDL, France, 2004.

[17] K. L. Man, *$SystemC^{\mathbb{FL}}$: Formalization of SystemC*, in IEEE Proceedings of the 12th Mediterranean Electrotechnical Conference—MELECON 2004, Dubrovnik, Croatia, May, 2004.

[18] K. L. Man, *Formal Communication Semantics of $SystemC^{\mathbb{FL}}$*, in IEEE Proceedings of the 8th Euromicro Conference on Digital System Design—DSD05, Porto, Portugal, September, 2005.

[19] $SystemC^{\mathbb{FL}}$ *homepage*, http://digilander.libero.it/systemcfl/

[20] K. L. Man, M. Boubekeur, M. P. Schellekens, *Process Algebraic Approach to SystemVerilog*, in IEEE Proceedings of the 20th IEEE Canadian Conference on Electrical and Computer Engineering, Vancouver, British Columbia, Canada, April, 2007.

[21] M. R. Mousavi, *Structuring Structural Operational Semantics*, Ph. D. Thesis, Department of Computer Science, Eindhoven University of Technology, September 2005.

[22] L. Aceto, W. Fokkink, C. Verhoef, *Structural Operational Semantics*, in Bergstra et al. BPS01, pp. 197–292, 1999.

[23] *The SMV model checker and user manual*, are available at http://www-2.cs.cmu.edu/~modelcheck/

[24] G. J. Holzmann, *The SPIN Model Checker*, Primer and Reference Manual, Addison-Wesley, 2004.

[25] R. Alur, D. L. Dill, *A Theory of Timed Automata*, Theoretical Computer Science, Vol. 126, No. 2, pp. 183-236, April, 1994.

[26] K. G. Larsen, P. Pettersson, W. Yi, *UPPAAL in a Nutshell*, Journal of Software Tools for Technology Transfer (STTT), Vol 1, No. 1-2, pp. 134–152, 1997.

[27] M. P. Schellekens, R. Agarwal, A. Fedeli, Y. F. Lam, K. L. Man, M. Boubekeur, E. Popovici, *Towards Fast and Accurate Static Average-Case Performance Analysis of Embedded Systems: The $\mathcal{MOQA}$ Approach*, in IEEE Proceedings of the East-West Design and Test International Symposium, September, 2007.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Dealing with redundancies and dependencies in normalization of XML data

Tadeusz Pankowski
Institute of Control and Information Engineering
Poznań University of Technology
Pl. M.S.-Curie 5, 60-965 Poznań, Poland
Email: tadeusz.pankowski@put.poznan.pl

Tomasz Piłka
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
ul. Umultowska 87, 61-614 Poznań, Poland
Email: tomasz.pilka@amu.edu.pl

***Abstract*—In this paper we discuss the problem of redundancies and data dependencies in XML data while an XML schema is to be normalized. Normalization is one of the main tasks in relational database design, where 3NF or BCNF, is to be reached. However, neither of them is ideal: 3NF preserves dependencies but may not always eliminate redundancies, BCNF on the contrary—always eliminates redundancies but may not preserve constraints. We discuss the possibility of achieving both redundancy-free and dependency preserving form of XML schema. We show how the XML normal form can be obtained for a class of XML schemas and a class of XML functional dependencies.**

## I. Introduction

As XML becomes popular as the standard data model for storing and interchanging data on the Web and more companies adopt XML as the primary data model for storing information, XML schema design has become an increasingly important issue. Central objectives of good schema design is to avoid data redundancies and to preserve dependencies enforced by the application domain. Existence of redundancy can lead not only to a higher data storage cost but also to increased costs for data transfer and data manipulation. It can also lead to update anomalies.

One strategy to avoid data redundancies is to design redundancy-free schema. One can start from an intuitively correct XML schema and a given set of functional dependencies reflecting some rules existing in application domain. Then the schema is normalized, i.e. restructured, in such a way that the newly obtained schema has no redundancy, preserves all data (is a lossless decomposition) and preserves all dependencies. In general, obtaining all of these three objectives is not always possible, as was shown for relational schemas [1]. However, in the case of XML schema, especially thanks to its hierarchical structure, this goal can be more often achieved [2].

The normalization process for relational schemas was proposed in the early 70s by Codd [3], [4]. Then a number of different normal forms was proposed, where the most important of them such as 2NF, 3NF [3], BCNF [4], and 4NF [5] are discussed today in every database textbook [1], [6], [7]. These normal forms together with normalization algorithms, aim to deal with the design of relational database taking into account different types of data dependencies [8], [7]. The result of the normalization should be a well-designed database [9]. The process of normalization of an XML schema is similar: we have to choose an appropriate XML schema for a given DTD and a set of data dependencies.

Recently, research on normalization of XML data was reported in papers by Arenas and Libkin [10], [11], Kolahi and Libkin [2], [12], [13], Yu and Jagadish [14].

In this paper we discuss a method for normalizing a class of XML schemas into an XML normal form that is redundancy-free and preserves all XML functional dependencies. To this order we apply the theory proposed in [12] and [11]. The novelty of the paper is the following:

- We use a new language (preliminarily proposed in [15]) for expressing schemas in a form of tree-pattern formulas, and for specifying XML functional dependencies.
- We show how the proposed formalism can be used for normalizing XML schemas into normal form similar to that of BCNF with eliminating redundancies but preserving all functional dependencies.

The structure of the paper is the following. In Section 2 we introduce an running example and motivate the research. In Section 3 a relational form of the running example is considered and some problems with its normalization are discussed. Basic notations relevant to the discussed issue from the XML perspective, are introduced in Section 4. We define a notation for defining tree-pattern formulas and for specifying data dependencies: XML functional dependencies and keys. Next, in Section 5, we discussed different normalization alternatives—we show advantages and drawbacks of some schema choices. Finally, in Section 6, an XML normal form (XNF) is defined (according to [11]) and we show how this form can be reached for our running example. We discuss the problem from theoretical and practical points of view. Section 7 concludes the paper.

## II. Motivation—Redundancies in XML Data

In XML data, like in relational ones, redundancies are caused by bad design of schemas. There are two kinds of design problems [11]: first of them is caused by non-key functional dependencies and is typical for relational schema design, while the other is more closely related to the hierarchical structure of XML documents.

As we mentioned above, in the case of XML schemas some redundancy problems may also occur because of bad design of hierarchical structure of XML document.

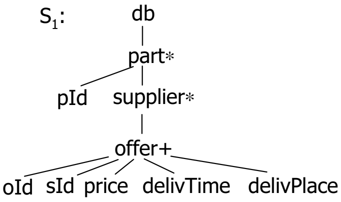


Fig. 1. Sample XML schema tree

```
db          →  part∗
part        →  pId supplier∗
pId         →  ϵ
supplier    →  offer+
offer       →  oId sId price delivTime delivPlace
oId         →  ϵ
sId         →  ϵ
price       →  ϵ
delivTime   →  ϵ
delivPlace  →  ϵ
```

Fig. 2. A DTD describing the XML schema in Fig. 1. The symbol $\epsilon$ denotes the empty string.

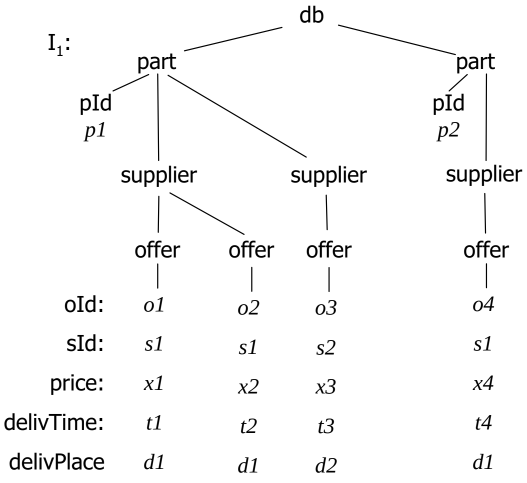


Fig. 3. Sample instance of schema $S_1$

*Example 2.1:* Let us consider the XML schema tree in Fig. 1 that describes a fragment of a database for storing data about parts and suppliers offering these parts. Its DTD specification is given in Fig. 2, and an instance of this schema is depicted in Fig. 3.

Each $part$ element has identifier $pId$. For each part may be zero or more suppliers offering this part. Offers are stored in $offer$ elements. Each offer has: offer identifier $oId$, supplier identifier $sId$, price $price$, delivery time $delivTime$, and delivery place $delivPlace$.

We assume that the following constraints must be satisfied by any instance of this schema:

- all $offer$ children of the same $supplier$ must have the same values of $sId$; this is similar to relational functional dependencies, but now we refer to both the values (text value of $sId$), and to structure (children of the same $supplier$).
- $delivPlace$ functionally depends on part ($pId$) and supplier ($sId$), i.e. when a supplier has two different offers for the same part (possibly with different $delivTime$ and/or $price$) the $delivPlace$ is the same - see offers $o1$ and $o2$ in Fig. 3.
- $delivPlace$ functionally determines supplier ($sId$). It means that having a delivery place ($delivPlace$) we know exactly which supplier is associated to this place; although one supplier can own many delivery places. For example, in Fig. 3 $d1$ is delivery place uniquely associated to the supplier $s1$.

It is easy to show that schema in Fig. 1 leads to redundancy: $sid$ (an also all other data describing suppliers such as e.g.: name and address) and $delivPlace$ are stored multiple times for a supplier.

Further on we will show that a special caution should be paid to such kind of dependencies as these in which participates $delivPlace$. In this case we have to do with "cyclic" dependencies, i.e. $delivPlace$ depends on $pId$ and $sId$ ($pId, sId \rightarrow delivPlace$) and $sId$ depends on $delivPlace$ ($delivPlace \rightarrow sId$).

First, we will discuss difficulties caused by such "cyclic" dependencies in relational databases, and next, we will show how this problem can be solved in the case of XML data.

## III. Dealing With Redundancies and Dependencies in Relational Databases

### A. Relational schemas and functional dependencies

In relational data model, a relational schema is understood as a pair

$$\mathcal{R} = (U, F),$$

where $U$ is a finite set of *attributes*, and $F$ is a set of *functional dependencies* over $F$. A functional dependence (FD) as an expression of the form

$$X \rightarrow Y,$$

where $X, Y \subseteq U$ are subsets of $U$. If $Y \subseteq X$, then $X \rightarrow Y$ is a *trivial* FD. By $F^+$ we denote all dependencies which can be inferred from $F$ by means of Armstrong's axioms [16], [7].

A *relation* of type $U$ is a finite set of tuples of type $U$. Let $U = \{A_1, ..., A_n\}$ and $dom(A)$ be the *domain* of attribute $A \in U$. Then a tuple $[A_1 : a_1, ..., A_n : a_n]$, where $a_i \in dom(A_i)$, is a tuple of type $U$.

A relation $R$ conforms to schema $\mathcal{R} = (U, F)$ (is an instance of this schema) if $R$ is of type $U$, and all dependencies from $F^+$ are satisfied by $R$. An FD $X \to Y$ is satisfied by $R$, denoted $R \models X \to Y$, if for all tuples $r_1, r_2 \in R$ holds

$$\pi_X(r_1) = \pi_X(r_2) \Rightarrow \pi_Y(r_1) = \pi_Y(r_2),$$

where $\pi_X(r)$ is the *restriction* (*projection*) of tuple $r$ on the set $X$ of attributes.

A *key* in $\mathcal{R} = (U, F)$ is such a minimal set $K$ of attributes that $K \to U$ is in $F^+$. Then each $A \in K$ is a *prime* attribute.

### B. Normalization of relational schemas

The main task in relational schema normalization is producing such a set of schemas that posses the required form, usually 3NF or BCNF. The normalization process consists in decomposition of a given input schema. The other approach consists in synthesizing 3NF from functional dependencies [8].

Ideally, a decomposition of a schema should be lossless, i.e. should preserve data and dependencies. Let $\mathcal{R} = (U, F)$, $U_1, U_2 \subseteq U$, and $U_1 \cup U_2 = U$, then schemas $\mathcal{R}_1 = (U_1, F_1)$ and $\mathcal{R}_2 = (U_2, F_2)$ are a lossless decomposition of $\mathcal{R} = (U, F)$, iff:

- The decomposition preserves data, i.e. for each instance $R$ of $\mathcal{R}$ the natural join of projections of $R$ on $U_1$ and $U_2$ produces the relation equal to $R$, i.e.
$$R = \pi_{U_1}(R) \bowtie \pi_{U_2}(R).$$
- The decomposition preserves dependencies, i.e.
$$F^+ = (F_1 \cup F_2)^+,$$
where $F_1 = \{X \to Y \mid X \to Y \in F \wedge X \cup Y \subseteq U_1\}$, and similarly for $F_2$.

The decomposition $((U_1, F_1), (U_2, F_2))$ of $(U, F)$ preserves data, if $U_1 \cap U_2 \to U_1 \in F^+$ (or, symmetrically, $U_1 \cap U_2 \to U_2 \in F^+$) [9], [7]. Then we say that the *decomposition is determined* by the functional dependence $U_1 \cap U_2 \to U_1 \in F^+$.

A schema $\mathcal{R} = (U, F)$ is in 3NF if for every FD $X \to A \in F^+$ holds:

- $X$ is a superkey, i.e. a key is a part of $X$, or
- $A$ is prime.

The second condition says that only prime attributes may be functionally dependent on a set of attributes which is not a key. A schema is in BCNF if only the first condition of the two above is allowed. It means, that if whenever a set $X$ determines functionally an attribute $A$, then $X$ is a superkey, i.e. determines the whole set $U$.

The aim of a normalization process is to develop normal forms by analyzing functional dependencies and successive decomposition of the input relational schema into its projections. This way a well-designed schema can be obtained, where unnecessary redundancies and update anomalies had been eliminated. In practice, 3NF is accepted as the most desirable form of relational schemas It does not eliminate all redundancies but guaranties dependency preservation. On contrast, BCNF eliminates all redundancies but does not preserve all dependencies. In [13] it was shown that 3NF has the least amount of redundancy among all dependency preserving normal forms. The research adopts a recently proposed information-theoretic framework for reasoning about database designs [10].

### C. Relational analysis of XML schema

Let us consider the relational representation of data structure presented in Fig. 1. Then we have the following relational schema:

$$\begin{aligned}
\mathcal{R} = \;& (U, F),\ \text{where} \\
U = \;& \{oId, sId, pId, price, delivTime, delivPlace\}, \\
F = \;& \{oId \to sId, pId, price, delivTime, delivPlace, \\
& \; sId, pId \to delivPlace, \\
& \; delivPlace \to sId\}.
\end{aligned}$$

In $\mathcal{R}$ there is only one *key*. The key consists of one attribute $oId$ since all attributes in $U$ functionally depends on $oId$. Thus, $R$ is in 2NF and $oId$ is the only prime (key) attribute in $\mathcal{R}$. Additionally, we assume that a given supplier delivers a given part exactly to one place ($pId, sId \to delivPlace$). Moreover, delivery place $delivPlace$ is connected with only one supplier ($delivPlace \to sId$).

$R$ is not in 3NF because for the functional dependency $sId, pId \to delivPlace$:

- $sId, pId$ is not a superkey, and
- $delivPlace$ is not a prime attribute in $U$.

Similarly for $delivPlace \to sId$.

The lack of 3NF is the source of redundancies and update anomalies. Indeed, for example, the value of $delivPlace$ will be repeated as many times as many different tuples with the same value of the pair $(sId, pId)$ exist in the instance of $\mathcal{R}$. To eliminate this drawback, we can decompose $\mathcal{R}$ into two relational schemas, $\mathcal{R}_1$ and $\mathcal{R}_2$, which are in 3NF. The decomposition must be based on the dependency $sId, pId \to delivPlace$ which guarantees that the decomposition preserves data. As a result, we obtain:

$$\begin{aligned}
\mathcal{R}_1 = \;& (U_1, F_1),\ \text{where} \\
U_1 = \;& \{oId, sId, pId, price, delivTime\}, \\
F_1 = \;& \{oId \to sId, pId, price, delivTime\}.
\end{aligned}$$

$$\begin{aligned}
\mathcal{R}_2 = \;& (U_2, F_2),\ \text{where} \\
U_2 = \;& \{sId, pId, delivPlace\}, \\
F_2 = \;& \{sId, pId \to delivPlace, \\
& \; delivPlace \to sId\}.
\end{aligned}$$

The discussed decomposition is both data and dependencies preserving, since:

$$R(U) = \pi_{U_1}(R) \bowtie \pi_{U_2}(R),$$

for every instance $R$ of schema $\mathcal{R}$, and

$$F = (F_1 \cup F_2)^+.$$

However, we see that $\mathcal{R}_2$ is not in BCNF, since $delivPlace$ is not a superkey in $\mathcal{R}_2$.

The lack of BCNF in $\mathcal{R}_2$ is the reason of redundancies. For example, in table $R_2$ we have as many duplicates of $sId$ as

$R_2$

| $sId$ | $pId$ | $delivPlace$ |
|---|---|---|
| s1 | p1 | d1 |
| s1 | p2 | d1 |
| s1 | p3 | d2 |
| s2 | p1 | d3 |

many tuples with the same value of $delivPlace$ exist in this table.

We can further decompose $\mathcal{R}_2$ into BCNF schemas $\mathcal{R}_{21}$ and $\mathcal{R}_{22}$, taking $delivPlace \rightarrow sId$ as the base for the decomposition. Then we obtain:

$$\begin{aligned}\mathcal{R}_{21} = &\ (U_{21}, F_{21}),\ \text{where}\\ U_{21} = &\ \{delivPlace, sid\},\\ F_{21} = &\ \{delivPlace \rightarrow sId\}.\end{aligned}$$

$$\begin{aligned}\mathcal{R}_{22} = &\ (U_{22}, F_{22}),\ \text{where}\\ U_{22} = &\ \{pId, delivPlace\},\\ F_{22} = &\ \emptyset.\end{aligned}$$

After applying this decomposition to $R_2$ we obtain tables $R_{21}$ and $R_{22}$:

$R_{21}$

| $sId$ | $delivPlace$ |
|---|---|
| s1 | d1 |
| s1 | d2 |
| s2 | d3 |

$R_{22}$

| $pId$ | $delivPlace$ |
|---|---|
| p1 | d1 |
| p2 | d1 |
| p3 | d2 |
| p1 | d3 |

This decomposition is information preserving, i.e.

$$R_2 = R_{21} \bowtie R_{22},$$

but does not preserve functional dependencies, i.e.

$$F_2 \neq (F_{21} \cup F_{22})^+ = F_{21}.$$

We can observe some negative consequences of the loss of functional dependencies in the result decomposition.

Assume that we insert the tuple $(p1, d2)$ into $R_{22}$. The tuple will be inserted because it does not violate any constrain imposed on $\mathcal{R}_{22}$. However, taking into account table $R_{21}$ we see that supplier $s1$ (determined by $d2$ in force of $delivPlace \rightarrow sId$) offers part $p1$ in the place $d1$. Thus, the considered insertion violates functional dependency $sId, pId \rightarrow delivPlace$ defined in $\mathcal{R}_2$.

The considered example shows that in the case of relational databases we are not able to completely eliminate redundancies and also preserve all functional dependencies. It turns out ([13]) that the best form for relation schema is 3NF, although some redundancies in tables having this form can still remain.

In next section we will show that the hierarchical structure of XML documents can be used to overcome some of the limitations of relational normal forms [11]. As it was shown in [12], there are decompositions of XML schemas that are both information and dependency preserving. In particular, we can obtain a form of XML schema that is equivalent to BCNF, i.e. eliminates all redundancies, and additionally preserves all XML functional dependencies.

## IV. XML SCHEMAS AND INSTANCES

Schemas for XML data are usually specified by DTD or XSD [17]. In this section we will define XML schema by means of *tree-pattern formulas* (TPF) [18], [15]. Furthermore, we do not consider attributes in XML trees since they can always be represented by elements. Schemas will be used to specify structures of *XML trees*. Some other properties of XML trees are defined as *schema constraints*.

*Definition 4.1:* Let $L$ be a set of *labels*, and $\mathbf{x}$ be a vector of variables. A schema TPF over $L$ and $\mathbf{x}$ is an expression conforming to the syntax:

$$\begin{aligned} S\ &::=\ /l[E]\\ E\ &::=\ l = x \mid l[E] \mid E \wedge ... \wedge E,\end{aligned}$$

where $l \in L$, and $x \in \mathbf{x}$. In order to indicate the set and ordering of variables in $S$ we will write $S(\mathbf{x})$.

□

*Example 4.1:* For the schema tree from Fig. 1, the schema TPF has the following form:

$$\begin{aligned} S_1 = \ &/db[part[pId = x_1 \wedge supplier[offer[\\ &oid = x_2 \wedge sid = x_3 \wedge price = x_4 \wedge\\ &delivTime = x_5 \wedge delivTime = x_6]]]].\end{aligned}$$

□

We see that a schema TPF has the following properties:

- reflects the tree structure of XML data,
- binds variables to paths in the schema tree,
- is a well-formed XPath predicate according to [19].

*Definition 4.2:* Let $S$ be a schema TPF over $\mathbf{x}$ and let an atom $l = x$ occur in $S$. Then the path $p$ starting in the root and ending in $l$ is called the type of the variable $x$, denoted $type_S(x) = p$.

□

The type of $x_1$ in $S_1$ is: $type_{S_1}(x_1) = /db/part/pId$.

An XML database consists of a set of XML data. We define XML data as an unordered rooted node-labeled tree (XML tree) over a set $L$ of labels, and a set $Str \cup \{\perp\}$ of strings and the distinguished null value $\perp$ (both strings and the null value, $\perp$, are used as values of text nodes).

*Definition 4.3:* An *XML tree* $I$ is a tuple $(r, N^e, N^t, child, \lambda, \nu)$, where:

- $r$ is a distinguished *root node*, $N^e$ is a finite set of *element nodes*, and $N^t$ is a finite set of *text nodes*;
- $child \subseteq (\{r\} \cup N^e) \times (N^e \cup N^t)$ – a relation introducing tree structure into the set $\{r\} \cup N^e \cup N^t$, where $r$ is the root, each element node has at least one child (which is an element or text node), text nodes are leaves;
- $\lambda : N^e \rightarrow L$ – a function labeling element nodes with names (labels);
- $\nu : N^t \rightarrow Str \cup \{\perp\}$ – a function labeling text nodes with *text values* from $Str$ or with the null value $\perp$.

□

It will be useful to perceive an XML tree $I$ with schema $S$ over tuple of variables $\mathbf{x}$, as a pair $(S, \Omega)$ (called a $description$), where $S$ is the schema TPF, and $\Omega$ is a set of valuations of variables in $\mathbf{x}$. A valuation $\omega \in \Omega$ is a function assigning values from $Str \cup \{\perp\}$ to variables in $\mathbf{x}$, i.e. $\omega \ : \ \mathbf{x} \rightarrow Str \cup \{\perp\}$.

*Example 4.2:* The instance $I_1$ in Figure 3 can be represented by the following description:

$I_1 := (S_1(x_1, x_2, x_3, x_4, x_5, x_6), \{(p1, o1, s1, x1, t1, d1),$
$(p1, o2, s1, x2, t2, d1), (p1, o3, s2, x3, t3, d2),$
$(p2, o4, s1, x4, t4, d1)\})$.

□

An XML tree $I$ satisfies a description $(S, \Omega)$, denoted $I \models (S, \Omega)$, if $I$ satisfies $(S, \omega)$ for every $\omega \in \Omega$, where this satisfaction is defined as follows:

*Definition 4.4:* Let $S$ be a schema TPF over $\mathbf{x}$, and $\omega$ be a valuation for variables in $\mathbf{x}$. An XML tree $I$ satisfies $S$ by valuation $\omega$, denoted $I \models (S, \omega)$, if the root $r$ of $I$ satisfies $S$ by valuation $\omega$, denoted $(I, r) \models (S, \omega)$, where:

1) $(I, r) \models (/l[E], \omega)$, iff $\exists n \in N^e \ child(r, n) \wedge (I, n) \models (l[E], \omega)$;
2) $(I, n) \models (l[E_1 \wedge ... \wedge E_k], \omega)$, iff $\lambda(n) = l$ and $\exists n_1, ..., n_k \in N^e(child(n, n_i) \wedge (I, n_i) \models (E_i, \omega))$ for $1 \leq i \leq k$;
3) $(I, n) \models (l = x, \omega)$, iff $\lambda(n) = l$ and $\exists n' \in N^t(child(n, n') \wedge \nu(n') = \omega(x))$.

□

A description $(S, \Omega)$ represents a class of $S$ instances with the same set of values (the same $\Omega$), since elements in instance trees can be grouped and nested in different ways.

For example, the XML tree in Fig. 4 satisfies two descriptions $(S_1, \Omega_1)$, and $(S_2, \Omega_2)$ where:

$$\begin{aligned} S_1 = \ & /A[B = x_1 \wedge C = x_2], \\ \Omega_1 = \ & \{(b, c1), (b, c2)\}; \\ \\ S_2 = \ & /A[B = x_1 \wedge C = x_2 \wedge D = x_3], \\ \Omega_2 = \ & \{(b, c1, d1), (b, c2, d1)\}. \end{aligned}$$

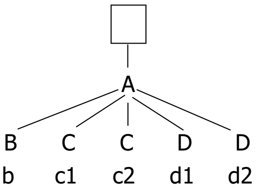


Fig. 4. A simple XML tree

## V. XML FUNCTIONAL DEPENDENCIES AND KEYS

Over a schema TPF we can define some constraints, which specify functional dependencies between values and/or nodes in instances of the schema. These constraints are called XML functional dependencies (XFD).

*Definition 5.1:* An XML functional dependency (XFD) over a set $L$ of *labels* and a set $\mathbf{x}$ of variables is an expression with the syntax:

$$\begin{aligned} fd \ & ::= \ /P[C]/.../P[C], \\ P \ & ::= \ l \mid P/l, \\ C \ & ::= \ TRUE \mid P = x \mid C \wedge ... \wedge C, \end{aligned} \tag{1}$$

where $l \in L$, and $x$ is a variable in $\mathbf{x}$. If variable names are significant, we will write $fd(\mathbf{x})$.

□

An XFD is an XPath expression that for a given valuation $\omega$ of its variables returns a sequence of objects (nodes or text values). An XML tree $I = (S, \Omega)$ satisfies an XFD $f(x_1, ..., x_k)$ if for each valuation $\omega \in \Omega$ of its variables, $f(x_1, ..., x_k)$ returns a singleton.

*Definition 5.2:* Let $I$ be an instance of schema TPF $S(\mathbf{x})$ and $f$ be an XFD defined over $S$. The instance $I$ satisfies $f$, denoted $I \models f$, if for every valuation $\omega$ of variables in $\mathbf{x}$, the implication holds

$$I \models (S, \omega) \Rightarrow count([\![f(\omega)]\!]) \leq 1,$$

where $[\![f(\omega)]\!]$ is the result of $f$ computed by the valuation $\omega$.

□

In the following example we discuss some XFDs over $S_1$.

*Example 5.1:* Over $S_1$ the following XFDs can be defined:

$$\begin{aligned} f_1(x) \ & = \ /db/part[supplier/offer/oId = x], \\ f_2(x) \ & = \ /db/part[supplier/offer/oId = x]/pId, \\ f_3(x_1, x_2) \ & = \ /db/part[pId = x_1]/supplier/offer[ \\ & \quad\ sId = x_2]/delivPlace, \\ f_4(x) \ & = \ /db/part/supplier/offer[ \\ & \quad\ delivPlace = x]/sId, \\ f_5(x_1, x_2) \ & = \ /db/part[pId = x_1]/supplier[ \\ & \quad\ offer/sId = x_2]. \end{aligned}$$

According to XPath semantics [19] the expression $f_1(x)(\omega)$ is evaluated against the instance $I_1$ as follows: (1) first, a set of nodes of type $/db/part$ is chosen; (2) next, for each chosen node the predicate $[supplier/offer/oId = x]$ is tested, this predicate is true in a node $n$, if there exists a path of type $supplier/offer/oId$ in $I_1$ leading from $n$ to a text node with the value $\omega(x)$. We see that $count([\![f_1(x)(\omega)]\!])$ equals 1 for all four valuations satisfied by $I_1$, i.e. for $\omega_i = [x \mapsto o_i]$, $1 \leq i \leq 4$.

Similarly, execution of $f_2(x)(\omega)$ returns a text value of the path $/db/part/pId$, where from the set of nodes determined by $/db/part$ are taken only nodes satisfying the predicate $[supplier/offer/oId = \omega(x)]$. We see that also this XFD is satisfied by $I_1$.

However, none of the following XFDs is satisfied in $I_1$:

$$\begin{aligned} g_1(x) \ & = \ /db/part[supplier/offer/sId = x], \\ g_2(x) \ & = \ /db/part[pId = x]/supplier/offer/sId \\ g_3(x) \ & = \ /db/part[pId = x]/supplier/offer \\ g_4(x_1, x_2) \ & = \ /db/part[pId = x_1]/supplier/offer[ \\ & \quad\ sId = x_2], \\ g_5(x) \ & = \ /db/part/pId/supplier/offer[ \\ & \quad\ delivPlace = x]. \end{aligned}$$

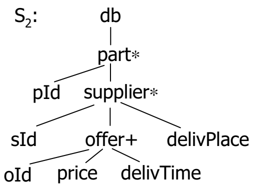


Fig. 5. Restructured form of schema in Fig. 1

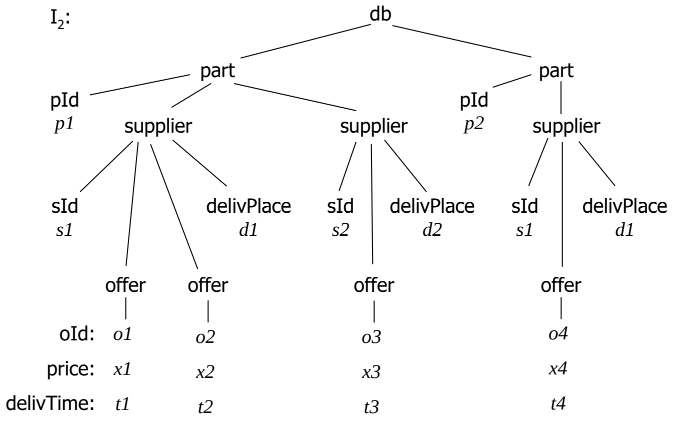


Fig. 6. Instance of schema in Fig. 5

Evaluating the above XFDs against $I_1$, we obtain:

$$\begin{array}{lcl} count([\![g_1(x)([x \mapsto s1])]\!]) & = & 2, \\ count([\![g_2(x)([x \mapsto p1])]\!]) & = & 2, \\ count([\![g_3(x)([x \mapsto p1])]\!]) & = & 3, \\ count([\![g_4(x_1,x_2)([x_1 \mapsto p1, x_2 \mapsto s1])]\!]) & = & 2, \\ count([\![g_5(x)([x \mapsto d1])]\!]) & = & 3. \end{array}$$

An XFD can determine functional relationship between a tuple of text values of a given tuple of paths and a path denoting either a text value (e.g. $f_2(x)$) or a subtree (a node being the root of the subtree) (e.g. $f_1(x)$).

There are also different notations to express functional notations. For example, according to the notation used in [11], the XFD $f_5(x_1, x_2)$ can be expressed as:

$$\begin{array}{l} db.part.@pId, db.part.supplier.offer.@sId \rightarrow \\ db.part.supplier, \end{array}$$

and using the language proposed in [20], as

$$\begin{array}{l} (db.part, \{pId\}), \\ (db.part, (supplier, \{offer/sId\})), \end{array}$$

where the first expression is an *absolute key* saying that $db.part$ is absolutely determined by $pId$, and the second expression is a *relative key* saying that in the context of $db.part$ a tree $supplier$ is determined by the path $offer.sId$.

Further on, by $(S, F)$ we will denote a schema TPF $S$ together with a set of XFDs defined over $S$. The closure of $(S, F)$, denoted by $(S, F)^+$, is the schema TPF $S$ and the set of all XFDs which can be derived from $(S, F)$.

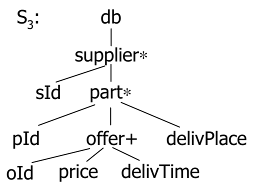


Fig. 7. Restructured form of schema in Fig. 1

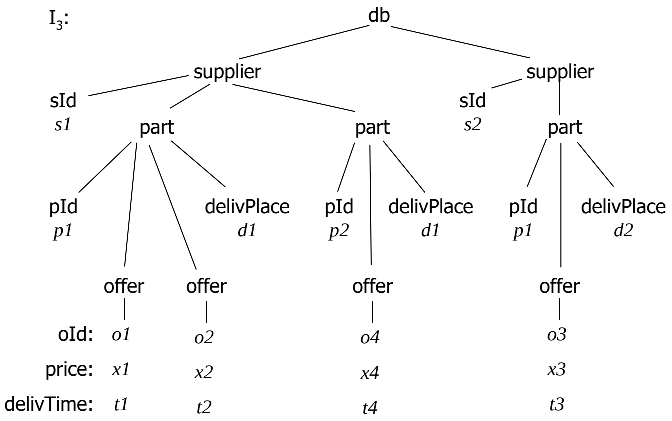


Fig. 8. Instance of schema in Fig. 7

## VI. Normal form for XML

To eliminate redundancies in XML documents, some normal forms for XML schemas have been proposed [21], [11], [2], [12]. We will define XNF (XML Normal Form) using tree-pattern formulas and functional dependencies defined in the previous section and adapting the approach proposed in [21].

*Definition 6.1:* Let $S$ be a schema TPF and $F$ be a set of functional dependencies over $S$. $(S, F)$ is in XML normal form (XNF) iff for every XFD $f/l \in (S, F)^+$, also $f \in (S, F)^+$, i.e.

$$(S, F) \text{ is in } XNF \text{ iff } f/l \in (S, F)^+ \Rightarrow f \in (S, F)^+.$$

□

Intuitively, let $f(x_1, ..., x_k)/l$ be XFD and $I$ be an instance of $S$. $I$ satisfies $f(x_1, ..., x_k)/l$, if for any valuation $\omega$ of the tuple of variables $(x_1, ..., x_k)$, there is at most one text value of the type $type(f/l)$. Thus, to avoid redundancies, for every valuation of $(x_1, ..., x_k)$ we should store the value of $f/l$ only once, i.e. there must be only one subtree of type $type(f)$ denoted by the expression $f(x_1, ..., x_k)$. In other words, XFD $f$ must be implied by $(S, F)$ [11].

Let us consider schema $S_2$ in Fig. 5 and its instance $I_2$ in Fig. 6. Then

$$f(x_1)/sId = /db/part/supplier[delivPlace = x_1]/sId$$

is XFD over $S_2$. This dependency corresponds to relational functional dependency $delivPlace \rightarrow sId$ and says that delivery place determines the supplier ($sId$).

However, $S_2$ is not in XNF, since its instance $I_2$ does not satisfy

$$f(x_1) = /db/part/supplier[delivPlace = x_1],$$

because for the valuation $\omega = [x_1 \mapsto "d1"]$ there are two different element nodes $supplier$ with value $d1$ of $delivPlace$. It means that $I_2$ is not free of redundancy.

In the case of schema $S_3$ (Fig. 7) the corresponding XFD has the form:

$$f(x_1)/sId = /db/supplier[part/delivPlace = x_1]/sId.$$

This dependency and also the XFD

$$f(x_1) = /db/supplier[part/delivPlace = x_1]$$

are satisfied by $I_3$ in Fig. 8. We see that this time for the valuation $\omega = [x_1 \mapsto "d1"]$, there is only one element node of type $supplier$ from which we can reach $delivPlace$ with value $d1$. This means that schema $S_3$ not only captures the XML functional dependency under consideration, but also is free of redundancies which may be caused by capturing this dependencies that happens in the case of schema $S_2$.

The other dependency of interest is $sId, pId \to delivPlace$. Its specification with respect to $S_2$ and $S_3$ is as follows:

$$/db/part[pId = x_1]/supplier[sId = x_2]/delivPlace,$$

and

$$/db/supplier[sId = x_1]/part[pId = x_2]/delivPlace.$$

It is easy to show that if these XFDs are satisfied by valuations, respectively, $\omega$ and $\omega'$ in instances of $S_2$ and $S_3$, then also

$$/db/part[pId = x_1]/supplier[sId = x_2],$$

and

$$/db/supplier[sId = x_1]/part[pId = x_2]$$

are satisfied by these valuations and these instances.

However, neither $S_2$ nor $S_3$ is in XNF. We have already shown that there is redundancy in instances of $S_2$. Similarly, we see that also in instances of $S_3$ redundancies may occur. Indeed, since one part may be delivered by many suppliers then the description of one part may be multiplied under each supplier delivering this part, so such data as *part name*, *type*, *manufacturer* etc. will be stored many times.

In Fig. 9 there is the final schema, $S_4$, for schemas under considerations: $S_1$, $S_2$, and $S_3$; $S_4$ is in XNF. To make the example more illustrative, we added node $name$ to $part$ data. Also the instance in Fig. 10 was slightly extended as compared with instances $I_2$ and $I_3$.

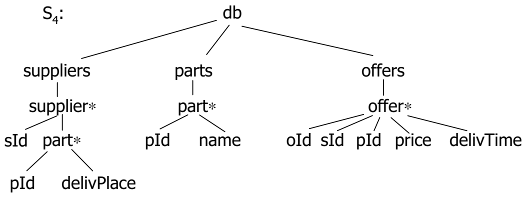


Fig. 9. XNF schema of schemas $S_1$, $S_2$, and $S_3$

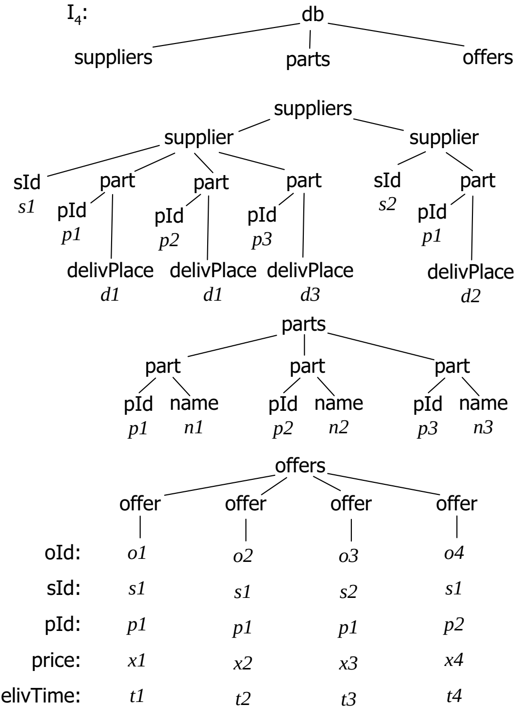


Fig. 10. Instance of schema in Fig. 9

XSD (XML Schema Definition) for $S_4$ in notation proposed in [17] is shown in Fig. 11.

$$\begin{array}{lll}
db & \to & \mathrm{db}[content] \\
content & \to & \mathrm{suppliers}[suppliers], \mathrm{parts}[parts], \\
 & & \quad \mathrm{offers}[offers] \\
suppliers & \to & \mathrm{supplier}[supplier]* \\
parts & \to & \mathrm{parts}[part_1]* \\
offers & \to & \mathrm{offers}[offer]* \\
supplier & \to & \mathrm{sId}[\epsilon], \mathrm{part}[part_2] \\
part_2 & \to & \mathrm{pId}[\epsilon], \mathrm{delivPlace}[\epsilon] \\
part_1 & \to & \mathrm{pId}[\epsilon], \mathrm{name}[\epsilon] \\
offer & \to & \mathrm{oId}[\epsilon], \mathrm{sId}[\epsilon], \\
 & & \quad \mathrm{pId}[\epsilon], \mathrm{price}[\epsilon], \mathrm{delivTime}[\epsilon]
\end{array}$$

Fig. 11. An XSD describing the XML schema in Fig. 9. The symbol $\epsilon$ denotes the empty string.

Note that we cannot use DTD since there are two subtrees labeled $part$, where each of them has different type: the $part$ subtree under $supplier$ consists of $pId$ and $delivPlace$, whereas the $part$ subtree under $parts$ consists of $pId$ and $name$. Recall that in the case of DTD each nonterminal symbol (label) can have only one type (definition), i.e. can appear on the left-hand side of exactly one production rule [17].

A set of XFDs for $S_4$ is defined in Fig. 12. XFDs derived from them are listed in Fig. 13.

$$\begin{array}{lcl}
f_1(x) & = & /db/suppliers/supplier[sId = x] \\
f_2(x_1, x_2) & = & /db/suppliers/supplier[sId = x]/ \\
 & & \quad part[pId = x_2]/delivPlace \\
f_3(x) & = & /db/suppliers/supplier[ \\
 & & \quad part/delivPlace = x]/sId \\
f_4(x) & = & /db/parts/part[pId = x]/name \\
f_5(x_1, x_2) & = & /db/offers/offer[sId = x_1 \wedge \\
 & & \quad pId = x_2]/oId \\
f_6(x) & = & /db/offers/offer[oId = x]/price \\
f_7(x) & = & /db/offers/offer[oId = x]/delivTime
\end{array}$$

Fig. 12. A set of XFD for the schema $S_4$

$$\begin{array}{lcl}
f'_2(x_1, x_2) & = & /db/suppliers/supplier[sId = x]/ \\
 & & \quad part[pId = x_2] \\
f'_3(x) & = & /db/suppliers/supplier[ \\
 & & \quad part/delivPlace = x] \\
f'_4(x) & = & /db/parts/part[pId = x] \\
f'_5(x_1, x_2) & = & /db/offers/offer[sId = x_1 \wedge pId = x_2] \\
f'_6(x) & = & /db/offers/offer[oId = x]
\end{array}$$

Fig. 13. A set of XFD derived from those in Fig. 12

We see that the schema $S_4$ satisfies the condition of XNF. Thus, this schema is both redundant-free and dependency preserving.

## VII. Conclusion

In this paper, we discussed how the concept of database normalization can be used in the case of XML data. Normalization is commonly used to develop a relational schema free of unnecessary redundancies and preserving all data dependencies existing in application domain. In order to apply this approach to design XML schemas, we introduced a language for expressing XML functional dependencies. In fact, this language is a class of XPath expressions, so its syntax and semantics are defined precisely. We define the notion of satisfaction of XML functional dependence by an XML tree. To define XNF we use the approach proposed in [11].

All considerations are illustrated by the running example. We discuss various issues connected with normalization and compare them with issues faced in the case of relational databases. We show how to develop redundancy-free and dependency preserving XML schema. It is worth mentioning that the relational version of the schema cannot be structured in redundancy-free and dependency preserving form. In this case, preservation of all dependencies requires 3NF but then some redundancy is present. Further normalization to BCNF eliminates redundancies but does not preserve dependencies. In the case of XML, thanks to its hierarchical nature, we can achieve both properties. However, it is not clear if this is true in all cases (see e.g. [12]).

In [15], [22], [23], XML functional dependencies (XFD) have been used in XML data integration settings, in particular for controlling query propagation in P2P environment and for reconciliation of inconsistent data.

**Acknowledgement:** This work was supported in part by the Polish Ministry of Science and Higher Education under grant 0014/R/2/T00/06/02.

## References

[1] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Reading, Massachusetts: Addison-Wesley, 1995.

[2] S. Kolahi, "Dependency-Preserving Normalization of Relational and XML Data," in *DBPL*, ser. Lecture Notes in Computer Science, G. M. Bierman and C. Koch, Eds., vol. 3774. Springer, 2005, pp. 247–261.

[3] E. Codd, "Further normalization of the data base relational model," *R. Rustin (ed.): Database Systems, Prentice Hall, and IBM Research Report RJ 909*, pp. 33–64, 1972.

[4] E. F. Codd, "Recent investigations in relational data base systems," in *IFIP Congress*, 1974, pp. 1017–1021.

[5] R. Fagin, "Multivalued dependencies and a new normal form for relational databases," *ACM Transactions on Database Systystems*, vol. 2, no. 3, pp. 262–278, 1977.

[6] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*. Redwood City: The Benjamin/Cummings, 1994.

[7] T. Pankowski, *Podstawy baz danych (in Polish, Fundamentals of databases)*. Warszawa: Wydawnictwo Naukowe PWN, 1992.

[8] P. A. Bernstein, "Synthesizing third normal form relations from functional dependencies," *ACM Transactions on Database Systtems*, vol. 1, no. 4, pp. 277–298, 1976.

[9] J. Rissanen, "Independent components of relations," *ACM Transactions on Database Systems*, vol. 2, no. 4, pp. 317–325, 1977.

[10] M. Arenas and L. Libkin, "An information-theoretic approach to normal forms for relational and XML data." *J. ACM*, vol. 52, no. 2, pp. 246–283, 2005.

[11] M. Arenas, "Normalization theory for XML," *SIGMOD Record*, vol. 35, no. 4, pp. 57–64, 2006.

[12] S. Kolahi, "Dependency-preserving normalization of relational and XML data," *Journal of Computer and System Sciences*, vol. 73, no. 4, pp. 636–647, 2007.

[13] S. Kolahi and L. Libkin, "On redundancy vs dependency preservation in normalization: an information-theoretic study of 3NF," in *PODS '06*. New York, NY, USA: ACM, 2006, pp. 114–123.

[14] C. Yu and H. V. Jagadish, "XML schema refinement through redundancy detection and normalization," *VLDB Journal*, vol. 17, no. 2, pp. 203–223, 2008.

[15] T. Pankowski, "XML data integration in SixP2P: a theoretical framework," in *EDBT Workshop on Data Management in P2P Systems DaMaP 2008*, ser. ACM International Conference Proceeding Series, A. Doucet, S. Gançarski, and E. Pacitti, Eds. ACM, 2008, pp. 11–18.

[16] W. W. Armstrong, "Dependency structures of data base relationships," in *IFIP Congress*, 1974, pp. 580–583.

[17] W. Martens, F. Neven, and T. Schwentick, "Simple off the shelf abstractions for XML schema," *SIGMOD Record*, vol. 36, no. 3, pp. 15–22, 2007.

[18] M. Arenas and L. Libkin, "XML Data Exchange: Consistency and Query Answering," in *PODS Conference*, 2005, pp. 13–24.

[19] XML Path Language (XPath) 2.0, 2006, www.w3.org/TR/xpath20.

[20] P. Buneman, S. B. Davidson, W. Fan, C. S. Hara, and W. C. Tan, "Reasoning about keys for XML," *Information Systems*, vol. 28, no. 8, pp. 1037–1063, 2003.

[21] M. Arenas and L. Libkin, "A normal form for XML documents." *ACM Trans. Database Syst.*, vol. 29, pp. 195–232, 2004.

[22] T. Pankowski, "Query propagation in a P2P data integration system in the presence of schema constraints," in *Data Management in Grid and P2P Systems (DEXA Globe'2008)*, vol. Lecture Notes in Computer Science **5187**, 2008, pp. 46–57.

[23] ——, "Reconciling inconsistent data in probabilistic XML data integration," in *British National Conference on Databases (BNCOD) 2008*, vol. Lecture Notes in Computer Science **5071**, 2008, pp. 75–86.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Separation of Crosscutting Concerns at the Design Level: an Extension to the UML Metamodel.

Adam Przybyłek
Gdańsk University, Department of Business Informatics,
Piaskowa 9, 81-824 Sopot, Poland
Email: adam@univ.gda.pl

***Abstract*—Aspect-oriented programming (AOP) was proposed as a way of improving the separation of concerns at the implementation level by introducing a new kind of modularization unit - an aspect. Aspects allow programmers to implement crosscutting concerns in a modular and well-localized way. As a result, the well-known phenomena of code tangling and scattering are avoided. After a decade of research, AOP has gained acceptance within both academia and industry. The current challenge is to incorporate aspect-oriented (AO) concepts into the software design phase. Since AOP is built on top of OOP, it seems natural to adapt UML to AO design. In this context the author introduces an extension to the UML metamodel to support aspect-oriented modelling.**

## I. Introduction

### *A. The evolution of the aspect-oriented paradigm*

THE TERM "crosscutting concern" describes part of a software system that should belong to a single module, but cannot be modularized because of the limited abstractions of the underlying programming language [20], [27], [33]. When crosscutting concerns are implemented using an object-oriented (OO) language, their code usually spreads over several core concerns [20], [26], [27]. Aspect-oriented programming (AOP) overcomes this problem by introducing a new unit of modularity—an aspect. Aspects allow programmers to avoid the well-known phenomena of code tangling and scattering, which adversely affect the readability, understandability, maintainability and reusability of the software [6], [20], [27], [30].

Programming and modelling languages exist in a relationship of mutual support. A software design co-ordinates well with a programming language when the abstraction mechanisms provided at both levels correspond to each other [26]. Successful adoption of AOP in both academia and industry has led to growing interest in aspect-oriented (AO) techniques for the whole software development lifecycle. Currently, one of the most active topics of research is modelling languages in support of aspect-orientation. Taking into account that (1) UML is considered to be the industry standard for OO system development and that (2) the AO paradigm complements the OO paradigm, it is quite natural to investigate UML as a possibility for the notation for aspect-oriented modelling (AOM) [2]–[4], [7], [18], [25], [28], [32], [34], [37]. Although UML was not designed to provide constructs to describe aspects, its flexible and extensible metamodel enables it to be adapted for domain-specific modelling [4], [23]. Thus in recent years a large number of proposals have been put forward in this area, but none of them has gained common acceptance. This paper is one more step towards closing the gap between AO concepts and UML.

### *B. The UML extensibility mechanisms*

There are two alternative methods of extending UML to incorporate aspects: by elaborating a Meta Object Facility [1] (MOF) metamodel or by constructing a UML profile. A UML profile is a predefined set of stereotypes, tagged values, constraints, and graphical icons which enable a specific domain to be modelled [1], [7], [9], [23], [30], [35]. It was defined to provide a light-weight extension mechanism [23], termed light-weight because it does not define new elements in the metamodel of UML. The intention of profiles is to give a straightforward mechanism for adapting the standard UML metamodel with constructs that are specific to a particular domain [23]. The advantages of choosing the light-weight extension mechanism are that models can be defined by applying a well-known notation and that generic UML tools can be used. On the other hand, the drawbacks are that, since stereotypes are extensions to the existing elements, certain principles of the original elements must be observed, and consequently expressiveness is constrained.

Elaborating an MOF metamodel is referred to as heavy-weight extension and is harder than constructing a profile. It also has far less tool support. However, the metamodel constructed can be as expressive as required. Another drawback of the heavy-weight mechanism is the introduction of interdependency between specific versions of UML and its extensions. If UML changes in any way, its extensions may also have to change.

### *C. Motivation and goals*

In the last few years, research in to AOM has concentrated on providing UML profiles, while less attention has been given to constructing heavy-weight extensions. The common

[1] Meta Object Facility (MOF) is the Object Management Group (OMG) standard, specifying how to define, interchange and extend metamodels.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

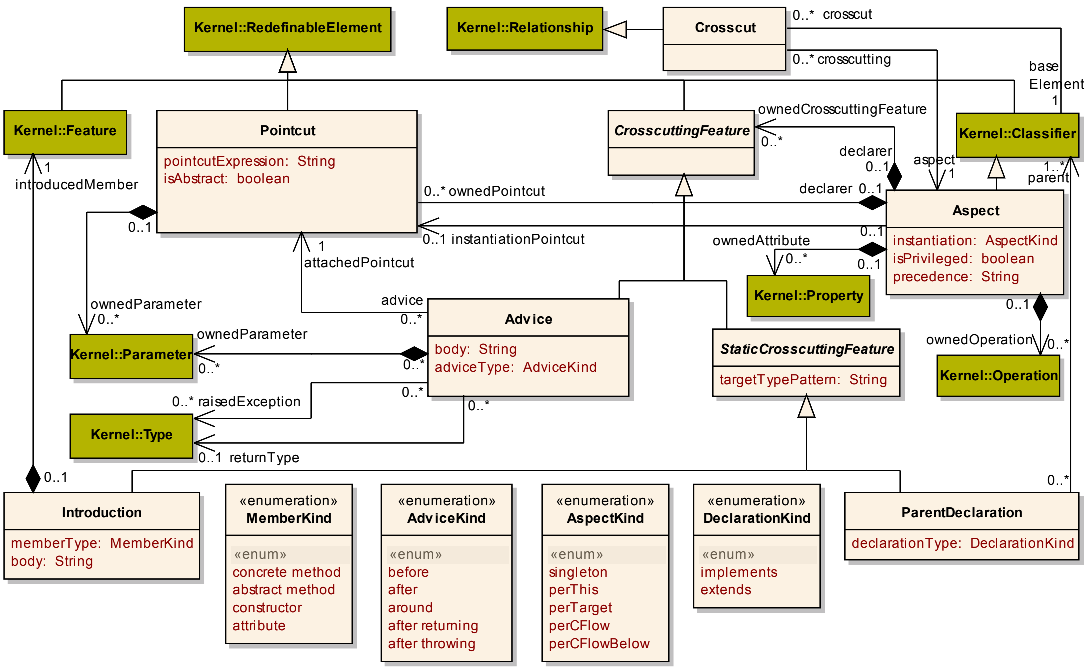


Fig. 1 . The AoUML package

practice [7], [9], [10] – [12], [22], [31], [32], [37] used to be to stereotype the class element as <<aspect>> and the method element as <<advice>>, although an aspect is not a class, nor is an advice a method. While such stereotyping was acceptable until UML 1.5, it can no longer be used; the 2.0 release requires semantic compatibility between a stereotyped element and the corresponding base element. In this context, using light-weight extensions is more an intermediate step in supporting the transition from OO modelling to AOM than a final solution.

The most valuable contributions to AOM have been made by Hachani [13], [14] and Yan [36], who proposed elaborately created and carefully specified metamodels for AspectJ. The main drawback of these extensions is the lack of graphical representation for new modelling elements. Moreover, they contain too much implementation detail and so seem to overwhelm the designer. Hachani's proposal is specified more strictly and in a more formal fashion but now needs updating, because it extends UML 1.4.

The motivation behind this research is to integrate the best practices of the existing AO extensions (particularly [5], [7], [13], [14], [16], [17], [19], [21], [29], [31], [32]) and to define a MOF metamodel that supplements the UML with means to AOM. The metamodel, which is presented in the next section, is based on the AspectJ approach to the AO paradigm. AspectJ has been chosen as the most representative AO programming language because of its mature implementation, industrial-strength tool support and wide popularity. Efforts [1], [8], [13], [28] to create a generic metamodel which could be fitted to each AO implementation have been unsuccessful, because a metamodel of this kind introduces an impedance mismatch between the design constructs and the language constructs. The conceptual differences between aspect implementations such as AspectJ, JAsCo, Spring, AspectWerkz are significant and cannot be captured effectively in a single metamodel. Moreover, generalizing aspects at the design level would be counter-productive at a time when AspectJ is squeezing out other technology at the implementation level.

## II. An extension to the UML metamodel

The elaborated extension is described by using a similar style to that of the UML metamodel. As such, the specification uses a combination of notations:

- UML class diagram – to show what constructs exist in the extension and how the constructs are built up in terms of the standard UML constructs;
- OCL – to establish well-formedness rules;
- natural language – to describe the semantic of the meta-classes introduced.

The proposed extension introduces a new package, named AoUML, which contains elements to represent the fundamental AO concepts of aspect, pointcut, advice, introduction, parent declaration and crosscutting dependency (Fig. 1).

The proposal reuses elements from the UML 2.1.2 infrastructure and superstructure specifications by importing the Kernel package. Fig. 2 shows the dependencies between the UML Infrastructure [23], the UML Superstructure [24] and the AoUML package.

### *A. Aspect meta-class*

#### *1) Semantic s*

An Aspect is a classifier that encapsulates the behaviour and structure of a crosscutting concern. It can, like a class, realize interfaces, extend classes and declare attributes and

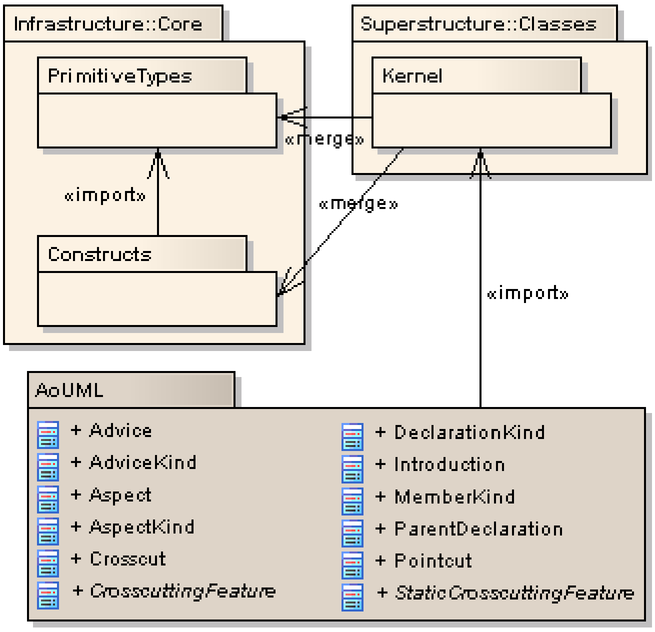


Fig. 2 . Dependencies between packages

operations. In addition, it can extend other aspects and declare advices, introductions and parent declarations.

*2) Attributes*

isPrivileged – if true, the aspect code is allowed to access private members of target elements as a "friend"; the default is false.

instantiation – specifies how the aspect is instantiated; the default is a singleton.

precedence – declares a precedence relationship between concrete aspects.

*3) Associations*

ownedPointcut – a set of pointcuts declared within the aspect.

instantiationPointcut – the pointcut which is associated with a per-clause instantiation model.

ownedCrosscuttingFeature – a set of crosscutting features owned by the aspect.

ownedAttribute – a set of attributes owned by the aspect.

ownedOperation – a set of operations owned by the aspect.

*4) Notation*

The aspect element looks similar to the class but has additional sections for pointcuts and crosscutting features declarations. Fig. 3 provides a graphical representation for an aspect.

### *B. CrosscuttingFeature meta-class*

*1) Semantic s*

A CrosscuttingFeature is an abstract meta-class, which declares a dynamic (an advice) or static feature to be combined to some target elements.

*2) Associations*

declarer – the aspect that owns this crosscutting feature.

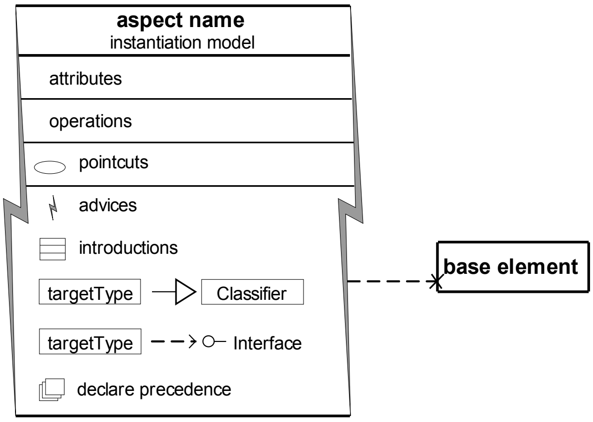


Fig. 3 . Aspect representation

### *C. StaticCrosscuttingFeature meta-class*

*1) Semantic s*

A StaticCrosscuttingFeature is a crosscutting feature that can be woven with core concerns on the basis of information available before runtime.

*2) Attributes*

targetTypePattern – a pattern that matches classes, interfaces or aspects which are affected by the crosscutting feature.

### *D. Introduction meta-class*

*1) Semantics*

An Introduction allows designers to add new attributes or methods to classes, interfaces or aspects.

*2) Attributes*

memberType – specifies the kind of the inter-type member declaration.

*3) Associations*

introducedMember – the new member which has to be added to the target type.

### *E. ParentDeclaration meta-class*

*1) Semantics*

A ParentDeclaration allows designers to add super-types to classes, interfaces or aspects.

*2) Attributes*

declarationType – specifies the kind of the declaration.

*3) Associations*

parent – the type implemented or extended by the target type.

### *F. Advice meta-class*

*1) Semantic s*

An Advice is a dynamic crosscutting feature that affects the behaviour of base classifiers. Each advice has exactly one associated pointcut and specifies the code that executes at each join-point picked out by the pointcut. The advice is able to access values in the execution context of the pointcut.

*2) Attributes*

adviceType – specifies when the advice code is executed relative to the join-points picked out.

body – the code of the advice.

*3) Associations*

ownedParameter – an ordered list of parameters to expose the execution context.

attachedPointcut – refers to the pointcut that defines a set of join-points at which the advice code is to be executed.

raisedException – a set of checked exceptions that may be raised during execution of the advice.

returnType – specifies the return result of the operation, if present (the "before" and "after" advice cannot return anything).

*4) Constraints* [2]

Advice parameters should have a unique name:

self.ownedParameter->forAll (p1, p2 | p1.name = p2.name implies p1=p2).

The before and after advice cannot return anything:

(self.adviceType = #before or self.adviceType = #after) implies (self.ownedParameter->forAll ( p | p.kind = #in)).

An advice can have at most one return parameter:

self.ownedParameter->
select (par | par.direction = #return)->size() <= 1.

### *G. Pointcut meta-class*

*1) Semantics*

A Pointcut is designed to specify a set of join-points and obtain the context surrounding the join-points as well. Join-points are well-defined places in the program flow where the associated advice must be executed. The purpose of declaring a pointcut is to share its pointcut expression in many advices or other pointcuts. A pointcut cannot be overloaded.

*2) Attributes*

isAbstract—if true, the Pointcut does not provide a complete declaration; the default value is false.

pointcutExpression—if a pointcut is not abstract, it specifies a set of join-points picked out by this pointcut; it has the same form as in AspectJ.

*3) Associations*

ownedParameter—an ordered list of parameters specifying what data is passed from runtime context to the associated advice.

advice—an advice that executes when the program reaches the join points.

*4) Notation*

The pointcut signature is as follows:

[visibility-modifier] pointcut name([parameters]): PointcutExpression

### *H. Crosscut meta-class*

*1) Semantics*

A Crosscut is a directed relationship, from an aspect to one or more base elements, where the additional structure and/or behaviour will be combined.

*2) Associations*

aspect – the aspect specifying the crosscutting concern affecting the base element.

baseElement – refers to the classifier that is crosscut .

## III. An Example

This section illustrates how the presented extension works in practice by modelling the Observer pattern adopted from Hannemann and Kiczales [15]. The participants in the Observer pattern are subjects and observers. The subject is a data structure which changes over time (such as a figure), and the observer (a screen) is an object whose own invariants depend on the state of the subject (Fig. 4).

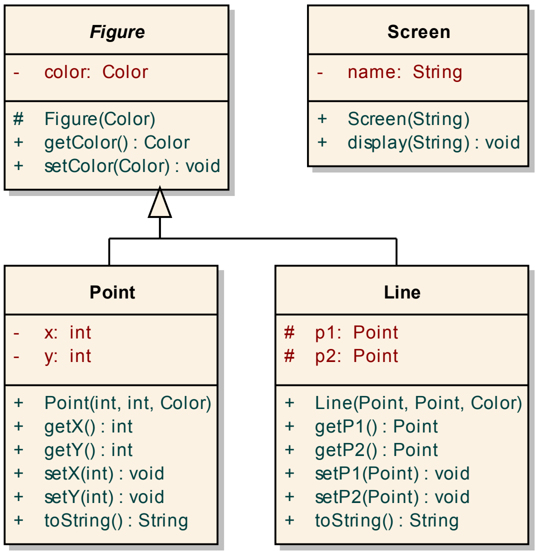


Fig. 4 . A typical scenario for the Observer pattern

The intent ion of the Observer pattern is to define a one-to-many dependency between a subject and multiple observers, so that when the subject changes state, all its observers are notified and updated automatically [15], [26]. The main problem with the OO implementation of this pattern is that it requires modification either to the structure of the classes that play the roles of Subjects and Observers or to the structure of the class hierarchy. It is therefore hard to apply the pattern to an existing design. Hanneman and Kiczales showed how the Observer pattern could effectively be implemented using AOP (Listing 1) [15].

To keep a figure display updated, the ColorObserver and PositionObserver aspects are introduced (Listing 2). Their after advices are triggered whenever a figure should be updated (the subjectChange pointcut is reached).

This paper shows how the Observer pattern could be specified using the AoUML extension. Fig. 5 gives a visual representation of Listing 1 and Listing 2.

## IV. Conclusion

The evolution of the AO paradigm is progressing from programming towards the design stage. Modularization of crosscutting concerns at the design phase should provide benefits in two areas: (1) the system model will be consistent with system implementation; (2) the artefacts developed will be more reusable and maintainable.

The contribution of this research is a MOF metamodel that enriches UML with constructs for modelling crosscutting

[2] Due to limitations on space, OCL constraints are not included for other elements.

```
public abstract aspect ObserverProtocol {
  protected interface Subject {};
  protected interface Observer {};
  private WeakHashMap perSubjectObservers;
  protected List getObservers(Subject s) {
    if (perSubjectObservers == null) perSubjectObservers = new WeakHashMap();
    List observers = (List)perSubjectObservers.get(s);
    if ( observers == null ) {
      observers = new LinkedList();
      perSubjectObservers.put(s, observers);
    }
    return observers;
  }
  public void addObserver(Subject s, Observer o) {
    getObservers(s).add(o);
  }
  public void removeObserver(Subject s, Observer o) {
    getObservers(s).remove(o);
  }
  protected abstract void updateObserver(Subject s, Observer o);

  protected abstract pointcut subjectChange(Subject s);
  after(Subject s): subjectChange(s) {
    Iterator iter = getObservers(s).iterator();
    while ( iter.hasNext() ) updateObserver(s, ((Observer)iter.next()));
  }
}
```

Listing 1. The AO implementation of the Observer pattern

concerns. Although many existing works on AOM either do not fit the UML standard or are not complete, there is some valuable research [7], [13], [14], [21], [36] which has inspired this work. Nevertheless, the presented research offers some advantages over these previous proposals. Firstly, the extension put forward is easier to comprehend for UML users than [14] and [21], while at the same time being powerful enough to express crosscutting concerns precisely.

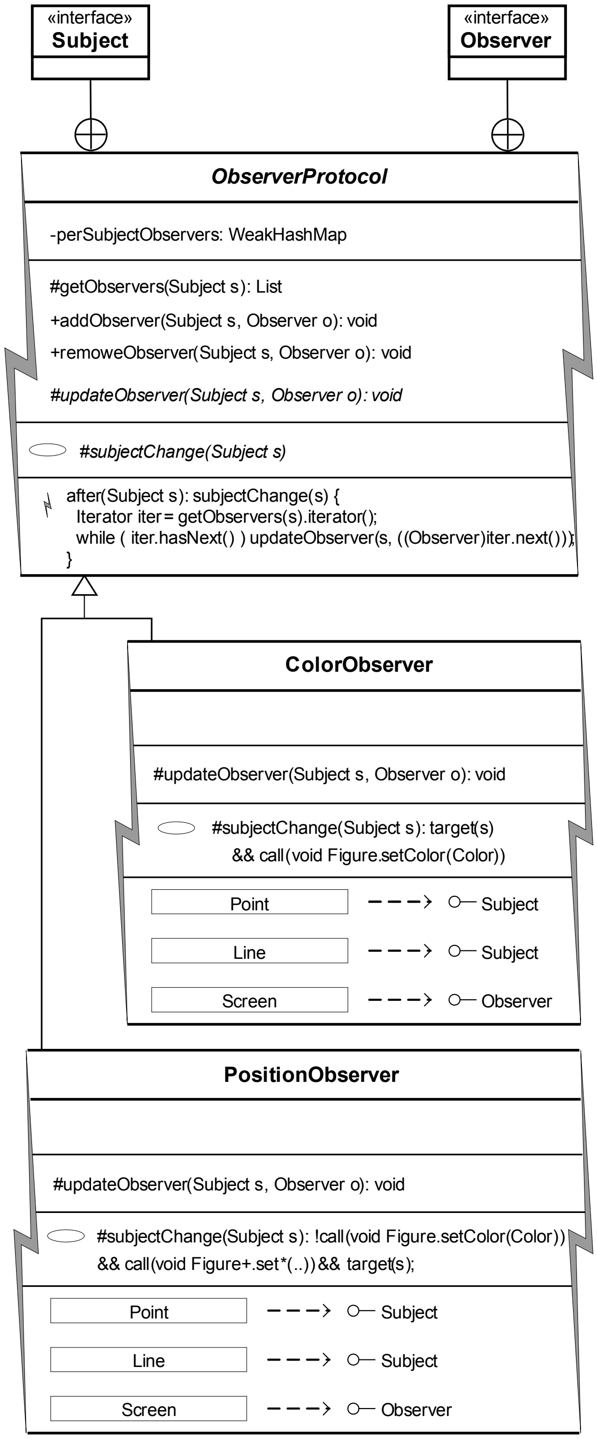
```
public aspect ColorObserver extends ObserverProtocol {
  protected void updateObserver(Subject s, Observer o) {
    ((Screen)o).display(s + " has changed the color");
  }

  protected pointcut subjectChange(Subject s):
    call(void Figure.setColor(Color)) && target(s);
  declare parents: Point implements Subject;
  declare parents: Line implements Subject;
  declare parents: Screen implements Observer;
}

public aspect PositionObserver extends ObserverProtocol {
  protected void updateObserver(Subject s, Observer o) {
    ((Screen)o).display(s + " has changed the position");
  }

  protected pointcut subjectChange(Subject s): target(s) &&
    !call(void Figure.setColor(Color)) && call(void Figure+.set*(..));
  declare parents: Point implements Subject;
  declare parents: Line implements Subject;
  declare parents: Screen implements Observer;
}
```

Listing 2. Definitions of two concrete observers


«interface»
Subject
«interface»
Observer
ObserverProtocol
-perSubjectObservers: WeakHashMap
#getObservers(Subject s): List
+addObserver(Subject s, Observer o): void
+remoweObserver(Subject s, Observer o): void
#updateObserver(Subject s, Observer o): void
#subjectChange(Subject s)
ColorObserver
#updateObserver(Subject s, Observer o): void
#subjectChange(Subject s): target(s)
&& call(void Figure.setColor(Color))
Point
Subject
Line
Subject
Screen
Observer
PositionObserver
#updateObserver(Subject s, Observer o): void
#subjectChange(Subject s): !call(void Figure.setColor(Color))
&& call(void Figure+.set*(..)) && target(s);
Point
Subject
Line
Subject
Screen
Observer

```
after(Subject s): subjectChange(s) {
  Iterator iter= getObservers(s).iterator();
  while ( iter.hasNext() ) updateObserver(s, ((Observer)iter.next()));
}
```

Fig. 5 . The class diagram using the AoUML extension

Moreover, in contrast to [7], [13], [14], [36] the presented metamodel provides dedicated icons for the aspect concepts. Graphical representation improves the understanding of models. Secondly, the proposal allows all aspect-related concepts to be specified in metamodel terms, so that no textual specification or notes are necessary. This means that automatic verification of the created models are simplified. Furthermore, the proposed metamodel does not modify the UML metamodel in any way; it merely adds some metaclasses. This contrasts with proposals that either are based on light-weight extensions [7] or modify the UML metamodel [13]. Thirdly, the extension put forward is adjusted to the newest UML specification (version 2.1.2). The main drawback of the proposal is that it has no support from the available modelling tools.

## References

[1] Aldawud, O., Elrad, T., Bader, A.: UML Profile for Aspect-Oriented Software Development. In: 3rd Workshop on Aspect-Oriented Modeling with UML at AOSD'03, Boston (2003)

[2] Barra, E., Genova, G., Llorens, J.: An Approach to Aspect Modelling with UML 2.0. In: 5th Aspect-Oriented Modeling Workshop at UML'04, Lisbon (2004)

[3] Basch, M., Sanchez, A.: Incorporating Aspects into the UML. In: 3rd Workshop on Aspect-Oriented Modeling with UML at AOSD'03, Boston (2003)

[4] Chavez, C., Lucena, C.: A Metamodel for Aspect-Oriented Modeling. In: Proceedings of the AOM with UML workshop at AOSD'02, Enschede (2002)

[5] Clarke, S., Banaissad, E.: Aspect-Oriented Analysis and Design: The Theme Approach. Addison-Wesley, Upper Saddle River (2005)

[6] Elrad, T., Aldawud, O., and Bader, A.: Aspect-Oriented Modeling: Bridging the Gap between Implementation and Design. In: 1st ACM SIGPLAN/SIGSOFT Conference on Generative Programming and Component Engineering (GPCE'02), Pittsburgh (2002)

[7] Evermann, J.: A meta-level specification and profile for AspectJ in UML. In: Journal of Object Technology, vol. 6(7), Special Issue: Aspect-Oriented Modeling, 27-49 (2007)

[8] France, R., Georg, G., Ray, I.: Supporting Multi-Dimensional Separation of Design Concerns. In: 3rd Workshop on Aspect-Oriented Modeling with UML at AOSD'03, Boston (2003)

[9] Fuentes, L., Sanchez, P.: Towards Executable Aspect-Oriented UML Models. In: 10th International Workshop on Aspect-Oriented Modeling at AOSD'07, Vancouver (2007)

[10] Gao, S., Deng, Y., Yu, H., He, X., Beznosov, K., Cooper, K.: Applying Aspect-Orientation in Designing Security Systems: a Case Study. In: 16th International Conference on Software Engineering (SEKE'04), Banff (2004)

[11] Groher, I., Baumgarth, T.: Aspect-Orientation from Design to Code. In: Workshop on Early Aspects at AOSD'03, Lancaster (2004)

[12] Groher, I., Schulze, S.: Generating Aspect Code from UML Models. In: 3rd Workshop on Aspect-Oriented Modeling with UML at AOSD'03, Boston (2003)

[13] Hachani, O.: Aspect/UML: extending UML metamodel for Aspect. Research report, France (2003)

[14] Hachani, O.: AspectJ/UML: extending UML metamodel for AspectJ. Research report, France (2003)

[15] Hannemann, J., Kiczales, G.: Design Pattern Implementation in Java and AspectJ. In 17th Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'02), Seattle (2002)

[16] Jacobson, I., Ng, P.: Aspect-Oriented Software Development with Use Cases. Addison-Wesley, Upper Saddle River (2005)

[17] Kande, M.M.: A Concern-Oriented Approach to Software Architecture. PhD. Swiss Federal Institute of Technology, Lausanne (2003)

[18] Kande, M.M., Kienzle, J., Strohmeier, A.: From AOP to UML - A Bottom-Up Approach. In: Proceedings of the AOM with UML workshop at AOSD'02, Enschede (2002)

[19] Kande, M.M., Kienzle, J., Strohmeier, A.: From AOP to UML: Towards an Aspect-Oriented Architectural Modeling Approach. Technical Report, Swiss Federal Institute of Technology Lausanne (2002)

[20] Kiczales, G. et.al.: Aspect-Oriented Programming. In: 11th European Conference on Object-Oriented Programming (ECOOP'97). LNCS, vol. 1241, pp. 220–242. Springer, New York (1997)

[21] Lions, J.M., Simoneau, D., Pilette, G., Moussa, I.: Extending Open-Tool/UML Using Metamodeling: an Aspect Oriented Programming Case Study. In: 2nd Workshop on Aspect-Oriented Modeling with UML at UML'02, Dresden (2002)

[22] Mosconi, M., Charfi, A., Svacina, J.: Applying and Evaluating AOM for Platform Independent Behavioral UML Models. In: 7th International Conference on Aspect-Oriented Software Development (AOSD'08), Brussels (2008)

[23] Object Management Group: OMG UML, Infrastructure, V2.1.2. Document Number: formal/2007-11-04, http://www.omg.org/spec/UML (2007)

[24] Object Management Group: OMG UML, Superstructure, V2.1.2. Document Number: formal/2007-11-02, http://www.omg.org/spec/UML (2007)

[25] Pawlak, R. et.al.: A UML Notation for Aspect-Oriented Software Design. In: Proceedings of the AOM with UML workshop at AOSD'02, Enschede (2002)

[26] Piveta, E.K., Zancanella, L.C.: Observer Pattern using Aspect-Oriented Programming. In: 3rd Latin American Conference on Pattern Languages of Programming, Porto de Galinhas (2003)

[27] Przybylek, A.: Post Object-Oriented Paradigms in Software Development: a Comparative Analysis. In: 1st Workshop on Advances in Programming Languages at IMCSI'07, Wisła (2007)

[28] Reina, A. M., Torres, J., Toro, M.: Towards Developing Generic Solutions with Aspects. In: 5th Aspect-Oriented Modeling Workshop at UML'04, Lisbon (2004)

[29] Sapir, N., Tyszberowicz, S., Yehudai, A.: Extending UML with Aspect Usage Constraints in the Analysis and Design Phases. In: 2nd Workshop on Aspect-Oriented Modeling with UML at UML'02, Dresden (2002)

[30] Schauerhuber, A. et.al.: A Survey on Web Modeling Approaches for Ubiquitous Web Applications. Technical Report, Vienna University of Technology, 2007

[31] Stein, D., Hanenberg, S., Unland, R.: An UML-based Aspect-Oriented Design Notation. In: Proceedings of the AOM with UML Workshop at AOSD'02, Enschede (2002)

[32] Stein, D., Hanenberg, S., Unland, R.: Designing Aspect-Oriented Crosscutting in UML. In: Proceedings of the AOM with UML Workshop at AOSD'02, Enschede (2002)

[33] Störzer, M., Hanneberg, S.: A Classification of Pointcut Language Constructs. In: SPLAT'05 Workshop, Chicago (2005)

[34] Suzuki, J., Yamamotto, Y.: Extending UML with Aspects: Aspect Support in the Design Phase. In: 3rd Aspect-Oriented Programming Workshop at ECOOP'99, Lisbon (1999)

[35] Wrycza, S., Marcinkowski, B., Wyrzykowski, K.: UML 2.0 in Information Systems Modeling. Helion, Warsaw (2005)

[36] Yan, H., Kniesel, G., Cremers, A.: A Meta Model and Modeling Notation for AspectJ. In: 5th Workshop on Aspect-Oriented Modeling at UML'04, Lisbon (2004)

[37] Zakaria, A. A., Hosny, H., Zeid, A.: A UML Extension for Modeling Aspect-Oriented Systems. In: 2nd Workshop on Aspect-Oriented Modeling with UML at UML'02, Dresden (2002)

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 559–566

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Distributed Internet Systems Modeling Using TCPNs

Tomasz Rak
Department of Computer and Control Engineering,
Rzeszow University of Technology,
Poland
Email: trak@prz-rzeszow.pl

Slawomir Samolej
Department of Computer and Control Engineering,
Rzeszow University of Technology,
Poland
Email: ssamolej@prz-rzeszow.pl

***Abstract*—This paper presents a Timed Coloured Petri Nets based programming tool that supports modeling and performance analysis of distributed World Wide Web environments. A distributed Internet system model, initially described in compliance with Queueing Theory (QT) rules, is mapped onto the Timed Coloured Petri Net (TCPN) structure by means of queueing system templates. Then, it is executed and analyzed using Design/CPN toolset. The proposed distributed Internet systems modeling and design methodology has been applied for evaluation of several system architectures under different external loads.**

## I. Introduction

ONE OF modern Internet (or Web) systems development approaches assumes that the systems consist of a set of distributed nodes. Dedicated groups of nodes are organized in layers (clusters) conducting predefined services (e.g. WWW service or data base service). Simultaneously, for a significant number of Internet applications some kind of soft real-time constraints are formulated. The applications should provide up-to-date data in set time frames [13]. The appearing of new abovementioned development paradigms cause that searching for a new method of modeling and timing performance evaluation of distributed Internet systems seems to be an up-to-date research path.

One of intensively investigated branch of Internet systems software engineering is formal languages application for modeling and performance analysis. Amid suggested solutions there are: algebraic description [7], mapping through Queueing Nets (QN) [4], [12], modeling using both Coloured Petri Nets (CPN) [8] and Queueing Petri Nets (QPN) [5].

Our approach proposed in this paper may be treated as extension of solutions introduced in [5]. Queueing Petri Nets (QPN) idea has been transferred onto formalism of Timed Coloured Petri Nets (TCPNs) [3]. To create classic queueing system models defined as in [2] we used Design/CPN tool package [1]. As a result we developed a programming tool which is able to map timed behavior of queueing nets by means of simulation. The Design/CPN performance tool [6] has been used to effectively capture and analyze data for created models. The main features of the preliminary version of our software tool were announced in [10].

The remaining work is organized as follows. In section 2, we introduce rules of mapping queueing systems into TCPNs. In the next section, we present a method of applying the TCPNs based queueing systems models (TCPNs templates) to distributed Internet system modeling. Section 4 focuses on results of simulation some detailed Internet system models while section 5 sums up the paper and includes our future research plans.

We assumed that the reader is familiar with TCPN formalism [3], [9] and with main features of Design/CPN tool [1], [6].

## II. Queueing System Implementation

*Queueing Net* usually consists of a set of connected *queueing systems*. Each *queueing system* is described by an arrival process, a waiting room and a service process. In the proposed programming tool, we worked out several TCPNs based *queueing system templates* (Processor Sharing (PS) and First In First Out (FIFO)) most frequently used to represent properties of distributed Internet system components. Each template represents a separate TCPN net (subpage) which may be included in the model of the system as a substitution transition (using hierarchical CP nets mechanisms [3]).

Queueing system properties are mapped to the TCPNs net as follows. At a certain level of system description, a part of hardware/software is modeled as a TCPN, where some dedicated *substitution transitions* are understand as queueing systems. To have the queueing functionality running ”under” selected transitions the mapping to adequate TCPNs subpages must be done. The corresponding subpages include the implementation of the adequate queueing system.

In fig. 1a a simple TCPN is presented where PS substitution transition is interpreted as a certain queueing system. The PS transition acquires the queueing system functionality when the subnet as in fig. 1b is substituted for it. Figure 1b illustrates an example queueing system -/M/1/PS/$\infty$ (exponential service times, single server, Processor Sharing service discipline and unlimited number of arrivals in the system; the queue’s arrival process in our modeling approach is defined outside of queueing system model). Packets to be served by given queueing system are delivered by port place `INPUT_PACKS`. Then, they are scheduled in a queue in `PACK_QUEUE` place. Every given time quantum (regulated by time multiset included in `TIMERS` place) the first element in the queue is selected to be served

978-83-60810-14-9/08/$25.00 © 2008 IEEE

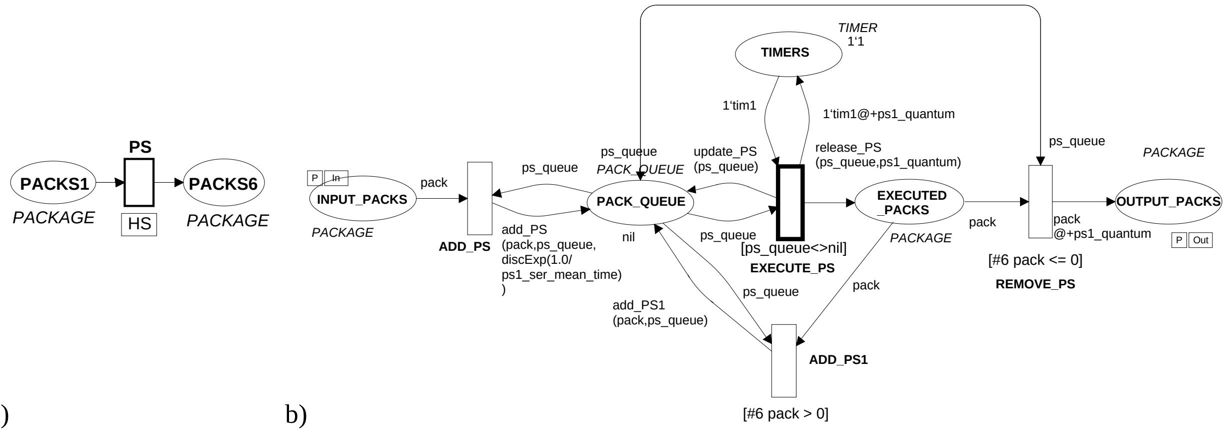


Fig. 1. TCPNs model of -/M/1/PS/∞ queuing system: a) primary model page b) detailed model page

(execution of transition EXECUTE_PS). Then, it is placed at the end of the queue or directed to leave the system (execution of transition ADD_PS1 or REMOVE_PS respectively). Number of tokens in TIMERS place represents number of servers for queueing system.

Full description of the model requires colors and functions definition in CPN ML language connected to the net elements:

```
val ps_ser_mean_time=1.0;
val pack_gen_mean_time=1.0;
color ID=int;    color PRT=int;
color START_TIME=int; color PROB=int;
color AUTIL=int; color RUTIL=int;
color INT=int; color TIMER=int timed;
color random_val=int with 1..100;
var tim_val:INT; var n:INT;
var tim1:TIMER; color PACKAGE=
   product ID*PRT*START_TIME*PROB*AUTIL*RUTIL timed;
var pack:PACKAGE; color PACK_QUEUE=list PACKAGE;
var ps_queue:PACK_QUEUE;
```

Corresponding arc functions (add_PS(), add_PS1(), update_PS(), release_PS()) release or insert tokens within the queue:

```
fun add_PS(pack:PACKAGE, queue:PACK_QUEUE,
ser_time:int)=if queue = nil
then [(#1 pack,#2 pack,#3 pack,#4 pack,
ser_time,ser_time)]
else (#1 pack,#2 pack,#3 pack,#4
pack,ser_time,ser_time)::queue;
fun add_PS1(pack:PACKAGE, queue:PACK_QUEUE)=
if queue=nil
then [pack] else pack::queue;
fun update_PS(queue:PACK_QUEUE)=rev(tl(rev queue));
fun release_PS(queue:PACK_QUEUE, ps_quantum:INT)=let
val r_pack=hd (rev queue) In
(#1 r_pack,#2 r_pack,#3 r_pack,ran'random_val(),
#5 r_pack, #6 r_pack-ps_quantum) end;
```

The state of the system is determined by the number and distribution of the tokens representing data packet flow. Each of the tokens representing a packet is a tuple PACKAGE = (ID, PRT, START_TIME, PROB, AUTIL, RUTIL) (compare source code including color's definitions), where: ID - token identification (allowing token class definition etc.), PRT - priority, START TIME - time of a token occurrence in the system, PROB - probability value (used in token movement distribution in the net), AUTIL - absolute value of token utilization factor (for PS queue) and RUTIL - relative value of token utilization factor. Tokens have *timed* attribute scheduling them within places which are not queues.

While packets are being served, the components of a tuple are being modified. At the moment the given packet leaves the queueing system, a new PROB field value of PACKAGE tuple is being generated randomly (release_PS function). The value may be used to modify the load of individual branches in the queueing system model. Generally, the queueing system template is characterized by the following parameters: average tokens service time (ps_ser_mean_time), number of servers (number of tokens in TIMERS place) and service discipline (the TCPN's structure).

In the software tool developed, it is possible to construct queueing nets with queueing systems having PS and FIFO disciplines. These disciplines are the most commonly used for modeling Internet systems. Some our previous works include the rules of mapping TCPNs into queues of tokens scheduled according priorities [9], [8]. The presented templates have been tested on their compatibility with mathematical formulas determining the average queue length and service time as in [2].

## III. Internet System Modeling and Analysis Approach

Having a set TCPN based queueing systems models a systematic methodology of Internet system modeling and analysis may be proposed. Typically, modern Internet systems are composed of layers where each layer consist of a set of servers—a server cluster. The layers are dedicated for proper tasks and exchange requests between each other.

To efficiently model typical Internet systems structures we proposed 3 modeling levels:

- superior—modeling of input process, transactions between layers and requests removal,
- layer—modeling of cluster structure,

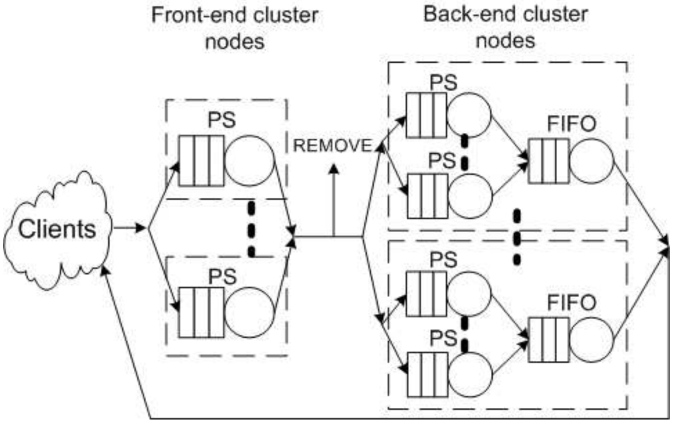


Fig. 2. Example distributed Internet system environment

- queue—modeling of queueing system.

To explain our approach to Internet system modeling a typical structure of distributed Internet system structure will be modeled and simulated. The example queueing model of the system consists of two layers of server clusters (fig. 2) and is constructed following the rules introduced in [4] and [5].

The front-end layer is responsible for presentation and processing of client requests. Nodes of this layer are modeled by PS queues. The next layer (back-end) implements system data handling. Nodes of this layer are modeled by using the serially connected PS and FIFO queues. The PS queue models the server processor and FIFO models the hard disc drive of server. Requests are sent to the system can be processed in both layers or removed after processing in front-end layer. The successfully processed requests are turned to the customer.

Figure 3 shows the TCPN based model of abovementioned queueing network. The superior level of system description is presented in fig. 3a, whereas in fig. 3b and 3c detailed queueing systems topologies at each layer of the system are shown (compare fig. 2). Server cluster of the first layer (e.g. WWW servers; fig. 3b) as well as the second layer cluster (e.g. database; fig. 3c) have been demonstrated on the main page of TCPN net as substituted transitions: `front-end_cluster` and `back-end_cluster`.

`T2` and `T3` transitions (compare fig. 3a) are in conflict. Execution of transition `T2` removes a token from the net (modeling the possible loss of data packet). However, if `T3` fires, the data packet is transferred for processing in the second layer of the system. Guard functions connected to the mentioned transitions determine proportions between the tokens (packets) rejected and the ones remaining in the queueing net (in the example model approximately `30%` of the tokens is rejected).

On the superior level of system description (fig. 3a) we have also defined arrival process of the queueing network (`T0` transition with `TIMER0` and `COUNTER` places). `TIMER0` place and `T0` transition constitute a clock-like structure that produces tokens (requests) according to random, exponentially distributed frequency. These tokens are accumulated in a form of timed multiset in `PACKS1` place and then forwarded into the queueing-based model of the Internet system. When each token is being generated its creation time is memorized in the `PACKAGE` tuple. This makes it possible to conduct an off-line analysis of the model.

Consequently, an executable (in a simulation sense) queueing network model is obtained. Tokens generated by arrival process are transferred in sequence by models of WWW server layer, by the part of the net that models loss (expiration) of some packets and by database layer. Provided that the system is balanced and has constant average arrival process, after some working time, the average values of the average queue length and response time are constant. Otherwise, their increase may occur.

The main parameters of the system modeled are the queue mean service time, the service time probability distribution function and the number of servicing units defined for each queueing system in the model. In the demonstrated model it has been assumed that queues belonging to a given layer have identical parameters.

At this stage of our research it has been decided that simulation will be the main mechanism used to do analysis of the constructed model. In our simulations we applied the performance analysis subsystem built in Design/CPN toolkit [6], [8]. It allows collecting selected elements of the net state at the moment of an occurrence certain events during simulation. It has been assumed that in each of the model layers, queue lengths and response time will be monitored. Monitoring of the abovementioned parameters helps to determine whether the model of the system is balanced. Fig. 4 shows example plots obtained in the simulation of the discussed model.

The example experiment covered model time range from `0` to `100 000` time units. Fig. 4a shows the state of selected queue when the modeled system was balanced. Response time does not increase and remains around average value. System is regarded as balanced if the average queue lengths in all layers do not increase. In fig. 4b response time for unbalanced system was shown. The results concern the same layer as previously and identical time range for the simulation. It is clear that response time (fig. 4b) increase during the experiment. On the basis of the plot in fig. 4b, it can be concluded that the modeled system under the assumed external load would be overload and probably appropriate modifications in the structure of the system would be necessary. The software tool introduced in our paper makes it possible to estimate the performance of developing Internet system, to test and finally to help adjust preliminary design assumptions.

Having the possibility to capture the net's state during the simulation within a certain time interval, it can be possible to select model parameters in such a manner that they meet assumed time restrictions. Additionally, the parameters of real Internet system can be used to fit parameters of the constructed model.

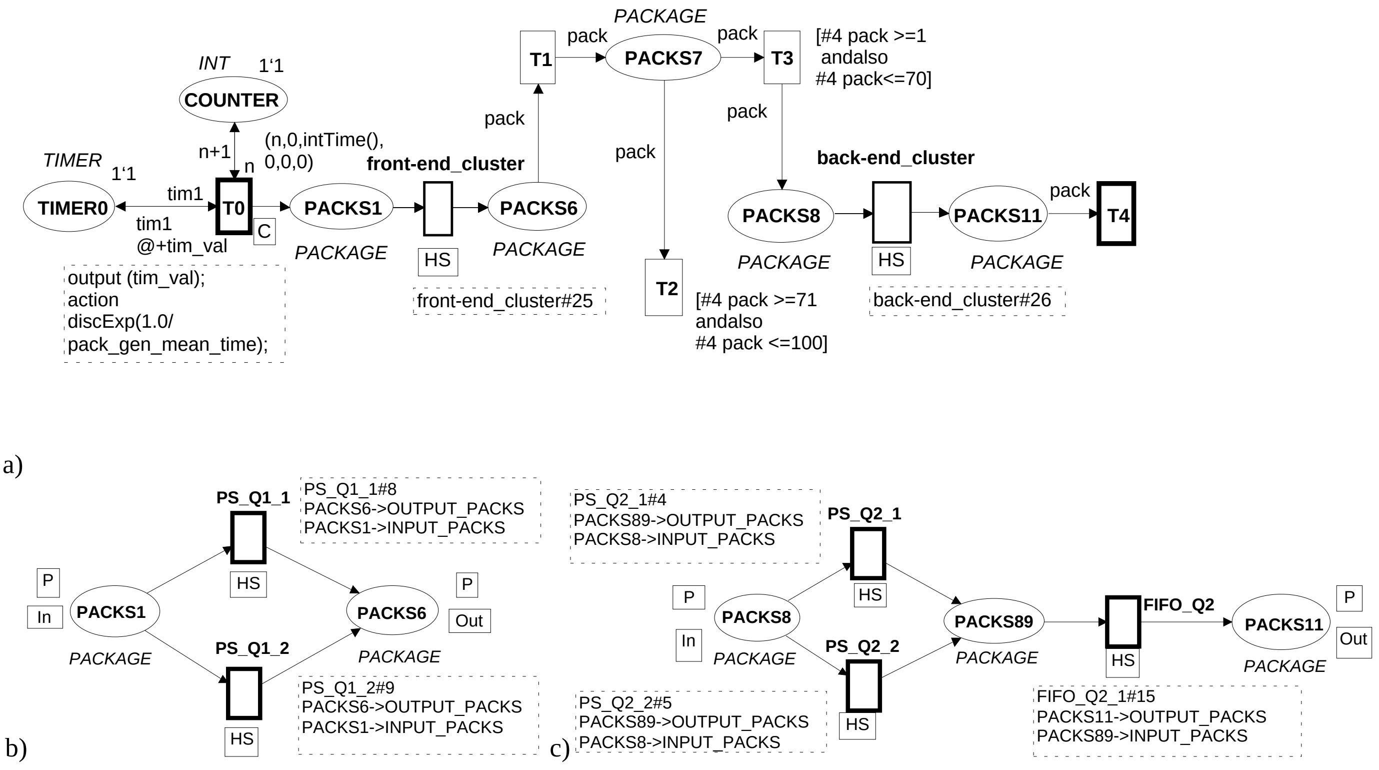


Fig. 3. TCPNs based queueing system model: a) main page, b) front-end_cluster subpage and c) back-end_cluster subpage

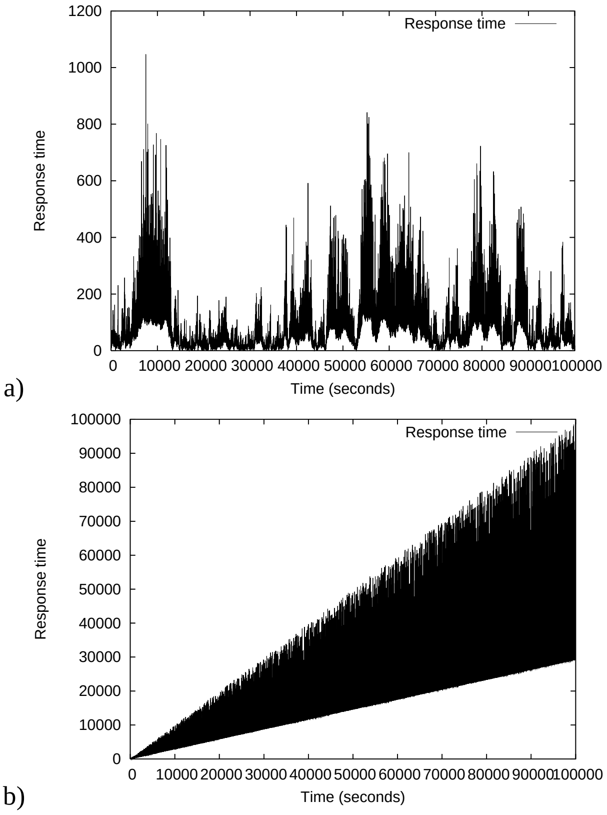


Fig. 4. Sample system response time history: a) system balanced and b) system unbalanced

## IV. Example System Model with Front-end Cluster and Back-end Replication

The worked out modeling and analysis methodology was used for construction and evaluation of several detailed models of architectures of distributed Internet systems. The analysis of the models were executed with use of performance analysis tools for TCPN nets [13]. CSIM [11] simulating environment and experiments on real Internet system were used for TCPNs simulations evaluation. The overview of typical TCPNs based Internet system models analyzed so far can be found in [8]. In the remaining part of the paper one example detailed model will be discussed: ”front-end cluster and back-end cluster with replication”.

### *A. Model description*

The queueing model of the example system is introduced in fig. 5. It consists of two cluster layers of servers. Let `A` be the number of homogeneous servers in the first and `B` in the second cluster layer respectively. Customer requests are sent to the chosen node of front-end cluster with `1/A` probability. Then they are placed in the queue to get service. The service in the service unit (processor) can be suspend many times, if for example the requests need the database access. When the database access occures, requests are sent to back-end layer. Any request can also be removed following `pREMOVE` path. In case of sending to the database, a requests steers itself to service in one of back-end nodes with `1/B` probability. The service in the database service unit may be suspend if access

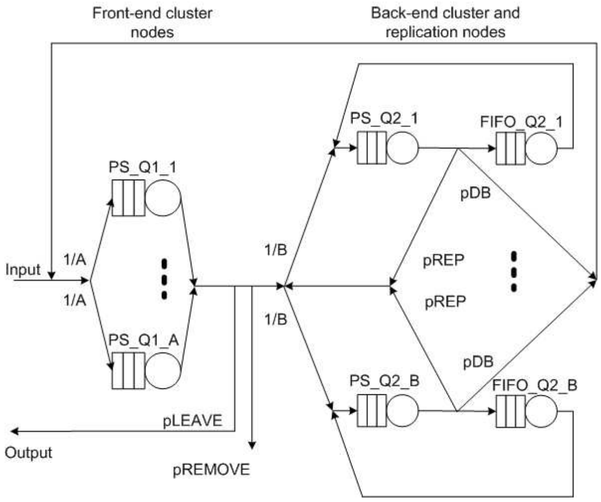


Fig. 5. Queueing model with cluster in front-end layer and replication in back-end layer

to the input/output subsystem of the database storage device is necessary. Requests are returned to the database service unit after the storage device is served. This operation can be repeated many times. After finishing servicing in the back-end layer, requests are returned to front-end servers (pDB). Requests can visit back-end layer during processing many times. After finish servicing in front-end server requests are sent to the customer (pLEAVE).

If the necessity of the database replication appears the requests are sent to next database node (pREP). Unless none of these two situations appear the requests are send to front-end layer (pDB). Replication can also cause resignation from transaction. Both the replication and the rejection of realizing transaction are modeled in simplistic way as delivery of task to the next location of replication.

In this model:

- PS_Q1_A is A queue PS modeling element that processes in front-end layer,
- PS_Q2_B is B queue PS modeling element that processes in back-end layer,
- FIFO_Q2_B is B queue FIFO modeling device that stores data in back-end layer.

Tab. I includes the probabilities values for Internet requests distribution for considered model. The parameters assumed for discussed model are as follows:

- external load,
- the identical parameters of queues,
- request distribution probabilities,
- the number of nodes in experimental environment.

In fig. 6a the main page (superior model level) of TCPN model that corresponds to the queueing model discussed above was presented. It makes it possible to transfer tokens from back-end layer back to front-end layer. It models the possibility of multiple tokens transfer to the server disk queue and replication. A subpage, which models back-end layer for database replication, is described on fig. 6b. The example shown contains two nodes of database (B=2) and its most essential properties are as following:

TABLE I
THE VALUE OF PROBABILITIES FOR MODEL

| Probability | Probability values for model [%] |
|---|---|
| pREMOVE | 30 |
| pLEAVE | 30 |
| pDB | 55 |
| pREP | 10 |

- the possibility of directing tokens to any node (the location of replication),
- the return of tokens at the beginning of the layer,
- the realization of data synchronisation in individual locations.

### B. Experimental and simulating model verification

Three example cases (configurations) of considered model were used to derive the architecture features. Individual cases mean as follows:

- case 1—A=2, B=2,
- case 2—A=4, B=2,
- case 3—A=4, B=4.

Experimental environment and the CSIM packet were used to verify proposed TCPN models. The experimental system consisted of a net segment (100Mb/s), set of computers (Pentium 4, 2.8 GHz, 256 MB RAM) with Linux operating system (kernel 2.4.22) and Apache2 software (for WWW servers) as well as MySQL, version 4.0.15 (for database servers) [8].

The verification model was written by using CSIM simulator. This is a process oriented discreet event simulation package used with C or C++ compilers [11]. It provides libraries that a program written can be used in order to model a system and to simulate it. The models created by using CSIM [8] were based on presented queue models (fig. 5) (similarly as TCPN models).

As a result we obtained the evaluated TCPNs based model of the Internet system discussed. The model made it possible to predict response time of the system developed. Average error between TCPN and CSIM models amounts to (tab. II) 9,5 % for response time. The comparison of results for TCPN models and experiments gave the following errors for individual model cases (tab. III) 14,9 %. In compare with experimental environment the average error of simulation for response time amounted to 15,1 % for TCPN and 14,1 % for CSIM respectively. In case of experiments (tab. III) there is the lack of compare for case 3 because of number of nodes in laboratory environment.

### C. Performance analysis

In fig. 7 the queue lengths for the model (cases 2 and 3) were shown. These cases differ from each other by a number of nodes in corresponding layers. Values of queue lengths

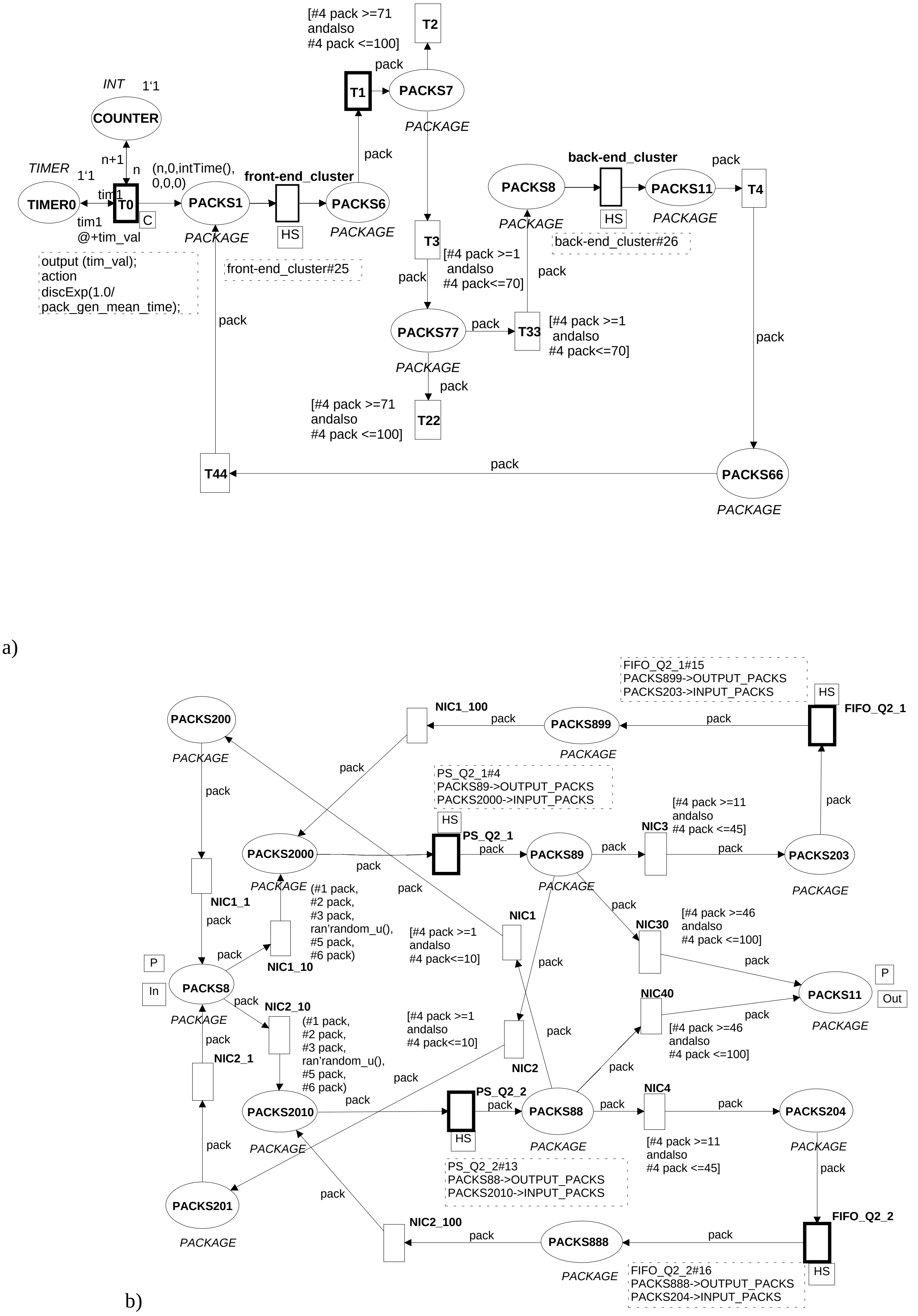


Fig. 6. TCPNs based queueing system model with cluster in front-end layer and replication in back-end layer: a) main page and b) back-end_cluster subpage

TABLE II
LAYERS RESPONSE TIME FOR TCPN MODEL AND CSIM MODEL

| Load [req./s] | Layer | TCPN Model [ms] | CSIM Model [ms] | Error [%] |
|---|---|---|---|---|
| Case 1 | | | | |
| 100 | front-end | 49 | 50 | -2.0 |
| 100 | back-end | 567 | 598 | -5.5 |
| 300 | front-end | 701 | 743 | -5.9 |
| 300 | back-end | 813 | 798 | 1.8 |
| 500 | front-end | 1001 | 1109 | -1.8 |
| 500 | back-end | 1050 | 1083 | -3.1 |
| Case 2 | | | | |
| 100 | front-end | 75 | 73 | 2.7 |
| 100 | back-end | 1083 | 867 | 19.9 |
| 300 | front-end | 79 | 85 | -7.6 |
| 300 | back-end | 1144 | 989 | 13.5 |
| 500 | front-end | 1042 | 1191 | -14.3 |
| 500 | back-end | 965 | 950 | -9.8 |
| Case 3 | | | | |
| 100 | front-end | 210 | 282 | -34.2 |
| 100 | back-end | 308 | 290 | 5.8 |
| 300 | front-end | 454 | 472 | -3.9 |
| 300 | back-end | 545 | 499 | 8.4 |
| 500 | front-end | 512 | 595 | -16.2 |
| 500 | back-end | 520 | 384 | 5.2 |

TABLE III
LAYERS RESPONSE TIME FOR TCPNS MODEL AND FOR EXPERIMENTAL REFERENCE SYSTEM

| Load [req./s] | Layer | TCPN Model [ms] | Experiments [ms] | Error [%] |
|---|---|---|---|---|
| Case 1 | | | | |
| 100 | front-end | 49 | 36 | 26.5 |
| 100 | back-end | 567 | 409 | 27.5 |
| 300 | front-end | 701 | 579 | 17.4 |
| 300 | back-end | 813 | 648 | 20.3 |
| 500 | front-end | 1001 | 921 | 8.0 |
| 500 | back-end | 1050 | 973 | 7.3 |
| Case 2 | | | | |
| 100 | front-end | 75 | 65 | 13.3 |
| 100 | back-end | 1083 | 791 | 26.9 |
| 300 | front-end | 79 | 79 | 0.0 |
| 300 | back-end | 1144 | 910 | 20.4 |
| 500 | front-end | 1042 | 975 | 6.4 |
| 500 | back-end | 965 | 925 | -6.9 |

in second layer were presented in charts (case 2—model for A=4, B=2): PSQ2_1 (fig. 7a) and PSQ2_2 (fig. 7b). Values of corresponding queue lengths for second layer in case 3 (A=4, B=4) were presented in charts: PSQ2_1 (fig. 7c) and PSQ2_2 (fig. 7d). Charts of both cases follow assumptions presented above and the same load. Queue lengths in case 2 are significantly longer than in case 3. It is easy to see benefits of enlarging number of nodes in back-end layer.

The performance problems were noticed during the distributed Internet systems analysis. Results of response time individual layers analysis (tab. II) illustrates behaviour observed by system customer. The growth of workload generally increases response time of the system. It was noticed that the use of clustering and nodes replication reduce the response time.

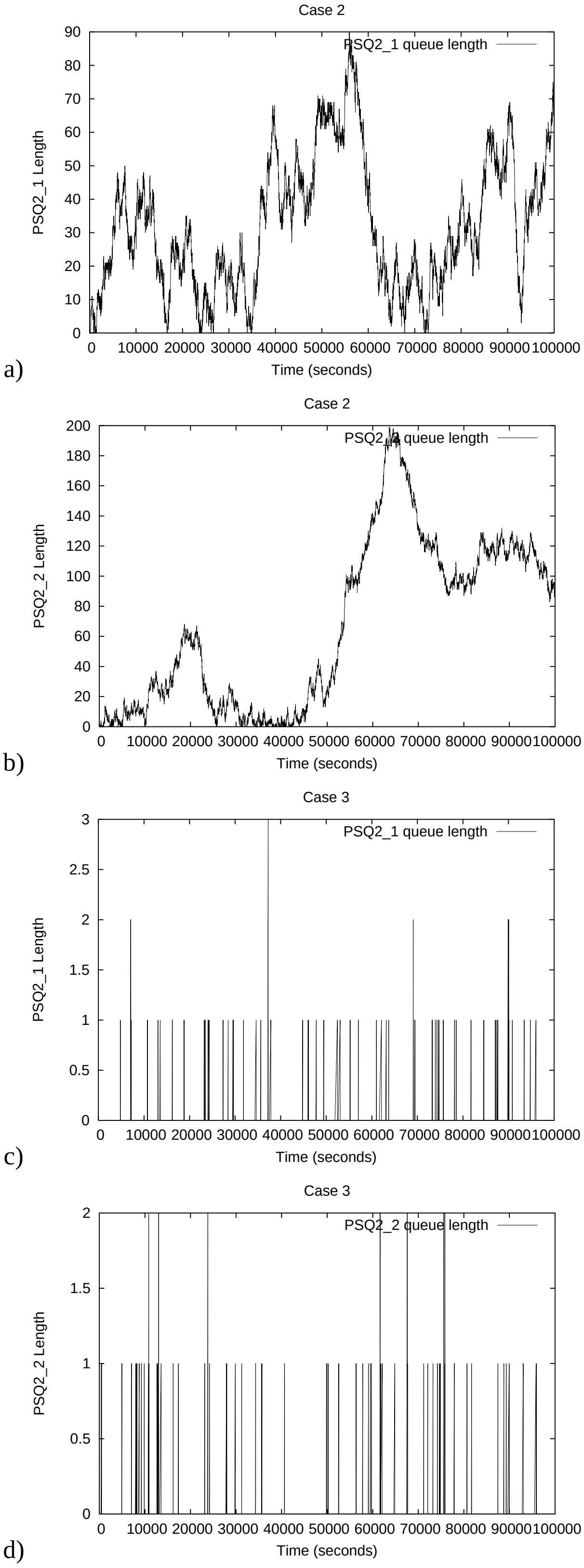


Fig. 7. PS queues lengths history in back-end layer: a), b) for case 2 and c), d) for case 3

In case of workload by more than 200 requests per second the system having the case 1 structure is overloaded. Backend layer becomes the system bottleneck. By the adequate modification of the number of computers in the second layer the overload condition under 200 request per second has been overcame. The presented performance analysis methodology makes it possible to detect early performance problems and to counteract them.

## V. Conclusion

It is still an open issue how to obtain an appropriate distributed Internet system. The demonstrated research results are an attempt to apply Queueing Theory (QT) and TCPNs formalism to the development of a software tool that can support distributed Internet system design. The idea of linking Queueing Nets Theory and Coloured Petri Nets was proposed previously by other authors in [5]. However, in the presented approach queueing systems have been implemented using TCPNs formalism exclusively. As a consequence, alternative implementation of Coloured Queueing Petri Nets has been proposed. What was more, the rules of modeling and analysis of distributed Internet systems applying described net structures was introduced.

This paper deals with the problem of calculating performance values like the response time in distributed Internet systems environment. The values are calculated by using the Design/CPN tool (TCPN). It is shown how the Coloured Petri net model of a distributed Internet system is created with some of its data structures and functions, and gives an examples of system analysis. A comparison of the results obtained by using the software tools (Design/CPN and CSIM) with the results acquired from the real system is presented.

The proposed approach is attempted to make a contribution to performance analysis of distributed Internet systems. This analysis is useful to determinate number and distribution of elements in the distributed Internet system architecture for specified requests load.

Our future research will focus on modeling and analyzing another structures of distributed Internet systems using the software tool developed. It will be also significant to demonstrate compatibility of the models with the real systems. TCPN features such as tokens distinction will be of more extensive use. We will also make an attempt to create queueing model systems with defined token classes and consider a possibility to use state space analysis of TCPN net to determine properties of the system.

## References

[1] Meta Software Corporation, *Design/CPN Reference Manual for X-Windows*, Meta Software, 1993.

[2] B. Filipowicz, *Modeling and optimize queueing systems*, POLDEX, Krakow, 2006. (In Polish)

[3] K. Jensen, *Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use*, Vol. 1, Springer, 1996.

[4] S. Konunev, A. Buchmann, *Performance Modeling and Evaluation of Large-Scale J2EE Applications*, In Proceedings of the 29th Int. Conf. of the Comp. Meas. Group on Res. Manag. and Perf. Eval. of Enterprise Comp. Syst., Dallas, Texas, December 7-12, pp. 486-502, 2003.

[5] S. Kounev, *Performance Engineering of Distributed Component-Base Systems, Banchmarking, Modeling and Performance Prediction*, Shaker Verlag, 2006.

[6] B. Linstrom, L. Wells, *Design/CPN Perf. Tool Manual*, CPN Group, Univ. of Aarhus, Denmark, 1999.

[7] T. Rak, *Model of Internet System Client Service*, Computer Science, Vol. 5, AGH Krakow, 55–65, 2003.

[8] T. Rak, *The Modeling and Analysis of Interactive Internet Systems Realizing the Service of High-Frequency Offers*, PhD dissertation supervised by J. Werewka, Krakow, AGH, 2007. (In Polish)

[9] S. Samolej, *Design of Embedded Systems Using Timed Coloured Petri Nets*, PhD dissertation supervised by T. Szmuc, Krakow, AGH, 2004. (In Polish)

[10] S. Samolej, T. Rak, *Time Properties of Internet Systems Modeling Using Coloured Petri Nets*, WKŁ, pp. 91–100, 2005. (In Polish)

[11] H. Schwetman, *CSIM19: A Powerfull Tool for Bilding System Models*, Proceedings Winter Simulation Conference, B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds., 2001.

[12] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, A. Tantawi, *An Analytical Model for Multi-tier Internet Service and Its Applications*, Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 291–302, 2005.

[13] L. Wells, S. Christensen, L. M. Kristensen, K. H. Mortensen, *Simulation Based Performance Analysis of Web Servers*, Proceedings of the 9th International Workshop on Petri Nets and Performance Models, IEEE, pp. 59–68, 2001.

# International Workshop on Real Time Software

INTERNATIONAL Workshop on Real Time Software will be held within the framework of the International Multiconference on Computer Science and Information Technology, and will be co-located with the XXIV Fall Meeting of Polish Information Processing Society.

Proliferation of computers interfacing with real world and controlling their environment requires careful investigation of approaches related to the specification, design, implementation, testing, and use of modern computer systems. Timing constraints, dependability, fault-tolerance, interfacing with the environment, reliability and safety constitute integral components of the software development process. Appropriate education of engineers developing such systems, working in interdisciplinary teams and in a global environment is of paramount importance.

In addition to traditional papers, we plan to organize a round table discussion forum on safety critical aspects of the education/training. We are soliciting brief one-page position papers on education and training of engineers developing dependable software intensive systems - presenting the views of academia and industry (in the submission clearly identity the "position paper" for the engineering education round table discussion). The accepted position papers will be a base for 10 minutes presentation followed by a discussion.

The workshop is planned around three main focus areas

- Real-Time Control
- Safety, Reliability, and Dependability
- Real-Time Education

Traditional Papers topics include but are not limited to:

- Real-time system development
- Scheduling
- Safety
- Reliability
- Dependability
- Fault-tolerance
- Feedback control real-time scheduling
- Hardware-software co-design
- Standards and certification
- Control software
- Robotics and UAV
- Software development tools
- Model-based development
- Automatic code generation
- Real-time systems education
- Related engineering curricula
- Laboratory infrastructure
- Internet-based support

## Steering Committee

**Andrew J. Kornecki** (Chairman), Embry Riddle Aeronautical University, USA

**Wojciech Grega,** AGH University of Science and Technology, Poland

**Janusz Zalewski,** Florida Gulf Coast University, USA

## Program Committee

**Marian Adamski,** University of Zielona Góra, Poland

**Karl-Erik Årzén,** Lund University, Sweden

**Mikhail Auguston,** Naval Postgraduate School, USA

**Jean-Philippe Babau,** INSA-Lyon, France

**Tuna Balkan,** Middle East Technical University, Turkey

**Manfred Broy,** Munich Technical University, Germany

**Matjaž Colnarič,** University of Maribor, Slovenia

**Alfons Crespo,** Universidad Politécnica de Valencia, Spain

**Albertas Čaplinskas,** Institute of Mathematics and Informatics, Lithuania

**Jindřich Černohorský,** VSB Technical University Ostrava, Czech Republic

**Karol Dobrovodský,** Slovak Academy of Sciences, Slovakia

**Frank Golatowski,** University of Rostock, Germany

**Luís Gomes,** Universidade Nova de Lisboa, Portugal

**Vladimir Hahanov,** Kharkov National University of Radio Electronics, Ukraine

**Wolfgang Halang,** FernUniversität Hagen, Germany

**Thomas Hilburn,** Embry Riddle Aeronautical University, USA

**Michael Hinchey,** Loyola College of Maryland, USA

**Janusz Kacprzyk,** Polish Academy of Sciences, Poland

**Wolfgang Kastner,** Vienna University of Technology, Austria

**Jan van Katwijk,** Delft University of Technology, The Netherlands

**Phil Laplante,** Penn State University, USA

**Tiberiu S. Letia,** Technical University of Cluj-Napoca, Romania

**György Lipovszki,** Budapest University of Technology and Economics, Hungary

**Anatiolijs Ļevčenkovs,** Riga Technical University, Latvia

**Jacek Malec,** Lund University, Sweden

**Gilles Motet,** INSA-Toulouse, France

**Leo Mõtus,** Estonian Academy of Sciences, Estonia

**Simin Nadjm-Tehrani,** Linköping University, Sweden

**Libero Nigro,** Università della Calabria, Italy

**Carlos E. Pereira,** Universidade Federal do Rio Grande do Sul, Brasil

**Rauf Sadykhov,** State University of Informatics and Radioelectronics, Belarus

**Bo I. Sandén,** Colorado Technical University, USA

**Ricardo Sanz,** Universidad Politecnica de Madrid, Spain

**Igor Schagaev,** London Metropolitan University, UK

**Vilém Srovnal,** VŠB Technical University Ostrava, Czech Republic

**Kari Systä,** Nokia Research Center, Finland

**Tomasz Szmuc,** AGH University of Science and Technology, Poland

**Miroslav Švéda,** Brno University of Technology, Czech Republic

**Jean-Marc Thiriet,** Laboratoire d'Automatique de Grenoble, France

**Leszek Trybus,** Politechnika Rzeszowska, Poland

**Andrzej Turnau,** AGH University of Science and Technology, Poland

**Shmuel Tyszberowicz,** Tel-Aviv University, Israel

**Tullio Vardanega,** Università di Padova, Italy

**Dieter Zöbel,** University Koblenz-Landau, Germany

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 569–575

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Real Time Behavior of Data in Distributed Embedded Systems

Tanguy Le Berre, Philippe Mauran, Gérard Padiou, Philippe Quéinnec
Université de Toulouse – IRIT
2, rue Charles CAMICHEL
31071 TOULOUSE Cedex 7
{tleberre,mauran,padiou,queinnec}@enseeiht.fr

***Abstract*—Nowadays, most embedded systems become distributed systems structured as a set of communicating components. Therefore, they display a less deterministic global behavior than centralized systems and their design and analysis must address both computation and communication scheduling in more complex configurations.**

**We propose a modeling framework centered on data. More precisely, the interactions between the data located in components are expressed in terms of a so-called observation relation. This abstraction is a relation between the values taken by two variables, the source and the image, where the image gets past values of the source. We extend this abstraction with time constraints in order to specify and analyze the availability of timely sound values.**

**The formal description of the observation-based computation model is stated using the formalisms of transition systems. Real time is introduced as a dedicated variable.**

**As a first result, this approach allows to focus on specifying time constraints attached to data and to postpone task and communication scheduling matters. At this level of abstraction, the designer has to specify time properties about the timeline of data such as their freshness, stability, latency…**

**As a second result, a verification of the global consistency of the specified system can be automatically performed. A forward or backward approach can be chosen. The verification process can start from either the timed properties (e.g. the period) of data inputs or the timed requirements of data outputs (e.g. the latency).**

**As a third result, communication protocols and task scheduling strategies can be derived as a refinement towards an actual implementation.**

## I. Introduction

DISTRIBUTED Real Time Embedded (DRE) systems become more and more widespread and complex. In this context, we propose a modeling framework centered on data to specify and analyze the real time behavior of these DRE systems. More precisely, such systems are structured as time-triggered communicating components. Instead of focusing on the specification and verification of time constraints upon computations structured as a set of tasks, we choose to consider data interactions between components. These interactions are expressed in terms of an abstraction called observation, which aims at expressing the impossibility for a site to maintain an instant knowledge of other sites. In this paper, we extend this observation with time constraints limiting the time shift induced by distribution. Starting from this modeling framework, the specification and verification of real time data behaviors can be carried out.

In a first step, we outline some related works which have adopted similar approaches but in different contexts and/or different formalisms.

Then, we describe the underlying formal system used to develop our distributed real time computation model, namely state transition systems. In this formal framework, we define a dedicated relation called observation to describe data interactions. An observation relation describes an invariant property between so-called “source” and “image” variables. Informally, at any execution point, the history of the image variable is a sub-history of the source variable. In other words, an observation abstracts the relation between arguments/results of a computation or between inputs/outputs of a communication protocol.

To express timed properties on the variables and their relation, we extend the framework so as to be able to describe the timeline of state variables. Therefore, for each state variable $x$, an abstraction of its timeline is introduced in terms of an auxiliary variable $\hat{x}$ which records its update instants. Then, real time constraints on data, for instance periodicity or steadiness, are expressed as differences between these dedicated variables and/or the current time. These auxiliary variables are also used to restrict the time shift between the source and the image of an observation. The semantics of the observation relation is extended to express different properties: for instance time lag or latency boundaries between the current value of the image and its corresponding source value.

The real time constraints about data behavior can be specified by means of these timed observations as illustrated in an automotive speed control example.

Lastly, we discuss the possibility to check the consistency of a specification. A specification is consistent if and only if the verification process can construct correct executions. However, the target systems are potentially infinite and an equivalent finite state transition system must be derived from the initial one before verification. The feasibility of this transformation is based upon assumptions about finite boundaries of the time constraints.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## II. State of the Art

We are interested in systems such as sensors networks. Our goal is to guarantee that the input data dispatched to processing units are timely sound despite the time shift introduced by distribution.

Most approaches taken to check timed properties of distributed systems are based on studying the timed behavior of tasks. For example, works like [1] propose to include the timed properties of communication in classical scheduling analysis.

Our approach is state-based and not event-based. We express the timed requirements as safety properties that must be satisfied in all states. The definition of these properties do not refer to the events of the system and is only based on the variable values. We depart from scheduling analysis by focusing on variable behavior and not considering the tasks and related system events.

Others approaches based on variables are mainly done in the field of databases. For example, the variables semantics and their timed validity domain are used in [2] to optimize transaction scheduling in databases. Our work is at a higher level as we propose to give an abstract description of the system in terms of a specification of data relations. Our framework can be used to check the correctness of an algorithm with regards to the freshness of the variables. It can also be used to specify a system without knowing its implementation.

Similar works are done using temporal logic to specify the system. For example, in [3], OCL constraints are used to define the validity domain of variables. A variation of TCTL is used to check the system synchronization and prevent a value from being used out of its validity domain. This work also defines timed constraints on the behavior and the relations between application variables, but these relations are defined using events such as message sending whereas our definitions are based on the variable values.

In [4], an Allen linear temporal logic is proposed to define constraints between intervals during which state variables remain stable. In other words, this approach also uses an abstraction of the data timelines in terms of stability intervals. However, the constraints remain logical and do not relate to real time. Nevertheless, this approach is intended to be applied in the context of autonomous embedded systems.

Using a semantics based on state transition system, we give a framework which aims at describing the relations between the data in a system, and specifying its timed properties and requirements.

## III. Theoretical Basis

### A. State Transition System

Models used in this paper are based on state transition systems. More precisely, this paper relies on the TLA+ formalism [5]. A *state* is an assignment of values to variables. A *transition relation* is a predicate on pair of states. A *transition system* is a couple (set of states, transition relation). A *step* is a pair of states which satisfies the transition relation. An *execution* $\sigma$ is any infinite sequence of states $\sigma_0\sigma_1\ldots\sigma_i\ldots$ such that two consecutive states form a step. We note $\sigma_i \rightarrow \sigma_{i+1}$ the step between the two consecutive states $\sigma_i$ and $\sigma_{i+1}$.

A *temporal predicate* is a predicate on executions; we note $\sigma \models P$ when an execution $\sigma$ satisfies the predicate $P$. Such a predicate is generally written in linear temporal logic. A *state expression* $e$ (in short, an expression) is a formula on variables; the value of $e$ in a state $\sigma_i$ is noted $e.\sigma_i$. The sequence of values taken by $e$ during an execution $\sigma$ is noted $e.\sigma$. A *state predicate* is a boolean-valued expression on states.

### B. Introducing Time

We consider real time properties of the system data. To distinguish them from (logical) temporal properties, such properties are called *timed* properties. Time is integrated in our transition system in a simple way, as described in [6]. Time is represented by a variable $T$ taking values in an infinite totally ordered set, such as $\mathbb{N}$ or $\mathbb{R}^+$. $T$ is an increasing and unbound variable. There is no condition on the density of time and moreover, it makes no difference whether time is continuous or discrete (see discussion in [7]). However, as an execution is a sequence of states, the actual sequence of values taken by $T$ during a given execution is necessarily discrete. This is the digital clock view of the real world. Note that we refer to the variable $T$ to study time and that we do not use the usual timed traces notation.

An execution can be seen as a sequence of snapshots of the system, each taken at some instant of time. We require that there are "enough" snapshots, that is that no variable can have different values at the same time and so in the same snapshot. Any change in the system implies time passing.

*Definition 1:* Separation. An execution $\sigma$ is separated if and only if for any variable $x$:

$$\forall i, j : T.\sigma_i = T.\sigma_j \Rightarrow x.\sigma_i = x.\sigma_j$$

In the following, we consider only separated executions. This allows to timestamp changes of variables and ensures a consistent computation model.

### C. Clocks

Let us consider a totally ordered set of values $\mathcal{D}$, such as $\mathbb{N}$ or $\mathbb{R}^+$. A clock is a (sub-)approximation of a sequence of $\mathcal{D}$ values. We note $[X \rightarrow Y]$ the set of functions with domain $X$ and range contained by $Y$.

*Definition 2:* A clock $c$ is a function in $[\mathcal{D} \rightarrow \mathcal{D}]$ such that:

- it never outgrows its argument value:
  $\forall t \in \mathcal{D} : c(t) \leq t$
- it is monotonously increasing:
  $\forall t, t' \in \mathcal{D} : t < t' \Rightarrow c(t) \leq c(t')$

In the following, clocks are used to characterize the timed behavior of variables. They are defined on the values taken by the time variable $T$, to express a time delayed behavior, as well as on the indices of the sequence of states, to express a logical precedence. A clock subset is used:

*Definition 3:* A clock $c$ from $[\mathcal{D} \rightarrow \mathcal{D}]$ is a *liveclock* if and only if:

$$\forall t \in \mathcal{D} : \exists t' \in \mathcal{D} : c(t') > c(t)$$

## IV. Specification of Data Timed Behavior

We introduce here the relation and properties used in our framework to describe the properties that must be satisfied by a system. Our approach is state-based and gives the relation that must be satisfied in all states. We define the observation relation to describe the relation between variables. A way to describe the timed behavior of the variables, that is properties of the history of data, is then introduced. We then extend the observation relation to take into account the timed constraints on the relation between variables. For that purpose we define predicates which bind relevant instants of the timeline of the source and the image of an observation. The predicates are expressed as bounds on the difference between two relevant instants.

### A. The Observation Relation

We define an observation relation on state transition systems as in [8]. The observation relation is used to abstract the value correlation between variables. The values taken by one variable are values previously taken by another variable.

Thus the observation relation binds two variables, the source $x$ and the image $`x$, and denotes that the history of the variable $`x$ is a sub-history of the variable $x$. The relation is defined by a couple $< \ source, image >$ and the existence of at least a clock that defines for each state which one of the previous values of the source is taken by the image. The formal definition is:

*Definition 4:* The variable $`x$ is an observation of the variable $x$ in execution $\sigma$: $\sigma \models `x \prec x$ iff:

$$\exists\ c \in [\mathbb{N} \to \mathbb{N}] : liveclock(c) \wedge\ \forall i : `x.\sigma_i = x.\sigma_{c(i)}$$

This relation states that any value of $`x$ is a previous value of $x$. Due to the properties of the observation clock $c$, $`x$ is assigned $x$ values respecting the chronological order. Moreover, $c$ always eventually increases, so $`x$ is always eventually updated with a new value of $x$. Figure 1 shows an example of an observation relation.

The observation can be used as an abstraction of communication in a distributed system, but it can as well be used as an abstraction of a computation:

- Communication consists in transferring the value of a local variable to a remote one. Communication time and lack of synchronization create a lag between the source and the image, which is modeled by $distant \prec local$.
- In state transition systems, a computation $f(x)$ is instantaneously computed. By writing $y \prec f(x)$, we model the fact that the computation takes time and the value of $y$ is based on the value of $x$ at the beginning of the computation.

In order to extend the observation relation with real time properties, we define instants that are used to characterize the timeline of variables.

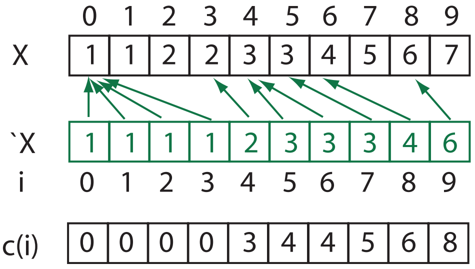


Fig. 1. The Observation Relation

### B. The Update Timeline of Variables

In order to state properties on the timed behavior of a variable $x$, we want to be able to refer to the last time this variable was updated. These are called the update instants $\hat{x}$. This referential can be either explicit or implicit. In the explicit case, the developer is responsible for giving its own variable $\hat{x}$. This can be the case if there is a periodic behavior of $x$ without having to describe actual values of $x$.

In the implicit case, a formal definition of $\hat{x}$ is given based on the history of the values taken by $x$. The goal is to capture the instant when the current value of $x$ appeared, e.g. the beginning of the current occurrence.

*Definition 5:* For a separated execution $\sigma$ and a variable $x$, the variable $\hat{x}$ is defined by:

$$\forall i : \hat{x}.\sigma_i = T.\sigma_{min\{j|\forall k\in[j..i]:\ x.\sigma_i = x.\sigma_k\}}$$

$\hat{x}$ is built from the history of $x$ values. For a variable $x$, the update instant of $x$ is defined as the value taken by the time $T$ at the earliest state when the current value appeared and continuously remained unchanged until the current state.

When $x$ is updated and its value changes then the value of $\hat{x}$ is also updated. Conversely if $\hat{x}$ changes then $x$ is updated. This property allows us to rely on the values of $\hat{x}$ to study the timed properties of $x$.

We also define the instant $Next(\hat{x})$ that returns the next value of $\hat{x}$ and thus the next instant when the value of $x$ is updated, i.e. the instant when the current value disappears. If $x$ is stable in a state $\sigma_i$ (no new update), then $Next(\hat{x}).\sigma_i = +\infty$.

### C. Variable Behavior

$\hat{x}$ is used to describe the timed behavior of a variable $x$. In this paper, we focus only on a certain type of variable. We expect each value of each variable to remain unchanged a bounded number of time units. We want to be able to give two characteristics for each variable: the minimum and the maximum duration between two updates. This basically describes two behaviors: a sporadic variable keeps each value for a minimum duration and, on the contrary, an alive variable has to be updated often, no value can be kept more than a given duration. These properties are expressed by a limit on the difference between $\hat{x}$ and $Next(\hat{x})$ using a characteristic called $Steadiness$. This bound denotes how long each value of $x$ can be kept.

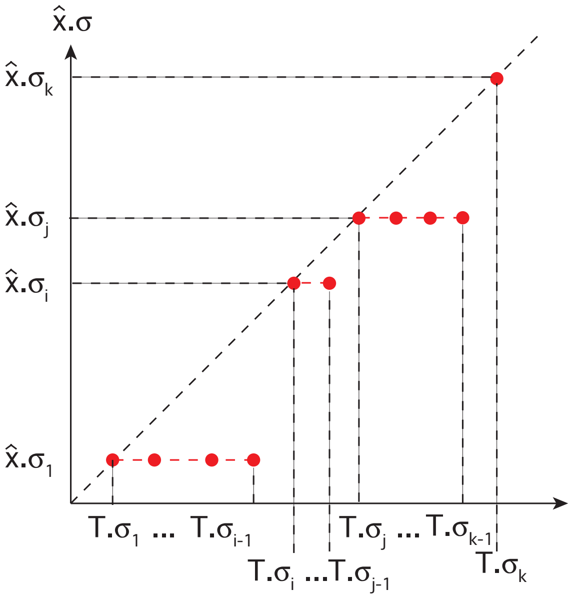


Fig. 2. Graph of $\hat{x}$

*Definition 6:* The steadiness of a variable $x$ is defined by:

$$\begin{array}{l}\sigma \models x\ \{Steadiness(\delta,\Delta)\} \triangleq\\ \qquad\qquad \forall i : \delta \leq Next(\hat{x}).\sigma_i - \hat{x}.\sigma_i < \Delta\end{array}$$

$\Delta - \delta$ is the jitter on $x$ updates. As long as there exists $\delta$ and $\Delta$ such that a variable timeline can be described using the $Steadiness$ feature then other properties can be given. For example, we introduce a stronger property, periodicity, where no time drift is allowed.

*Definition 7:* A variable $\hat{x}$ is periodic of period $P$ with jitter $J$ and phase $\phi$ iff:

$$\begin{array}{l}\sigma \models x\ \{Periodic(P,J,\Phi)\} \triangleq\\ \qquad\quad x\ \{Steadiness(P-2J,P+2J)\} \wedge\\ \forall i : \exists n \in \mathbb{N} : \hat{x}.\sigma_i \in [\phi + nP - J, \phi + nP + J]\end{array}$$

($J$ must verify $J < P/4$)
Such a variable is updated around all instants $\phi + nP$.

### D. Timed Observation

We use the update instants to extend the observation relation with timed characteristics. The timed constraints that extend the observation must capture the latency introduced by the observation and the modification of the timeline of the source to produce the timeline of the image. We define a set of predicates on the instants characterizing the source and the image timelines and the observation clock. Formally, a timed observation is defined as follows:

*Definition 8:* A timed observation is defined as an observation satisfying a set of predicates.

$$\begin{array}{l}\sigma \models {‘x} \prec x \left\{\begin{array}{l} Predicate_1(\delta_1,\Delta_1),\\ Predicate_2(\delta_2,\Delta_2),\\ \qquad\dots\end{array}\right\} \triangleq\\ \qquad\quad \exists c \in [\mathbb{N} \rightarrow \mathbb{N}] : liveclock(c) \wedge\\ \qquad\qquad\qquad \forall i : {‘x}.\sigma_i = x.\sigma_{c(i)} \wedge\\ \qquad\qquad\qquad Predicate_1(c,\delta_1,\Delta_1) \wedge\\ \qquad\qquad\qquad Predicate_2(c,\delta_2,\Delta_2)\dots\end{array}$$

The predicates that can be used to describe the timed properties of the relation between two variables are the following:

*Definition 9:* Given two variables ${‘x}$ and $x$ such that $\sigma \models {‘x} \prec x$ with a $liveclock$ $c \in [\mathbb{N} \rightarrow \mathbb{N}]$

$$\begin{array}{rcl} Lag(c,\delta,\Delta) & \triangleq & \delta \leq {‘\hat{x}}.\sigma_i - \hat{x}.\sigma_{c(i)} < \Delta\\ Stability(c,\delta,\Delta) & \triangleq & \delta \leq Next(\hat{x}).\sigma_{c(i)} - \hat{x}.\sigma_{c(i)} < \Delta\\ Latency(c,\delta,\Delta) & \triangleq & \delta \leq T.\sigma_i - \hat{x}.\sigma_{c(i)} < \Delta\\ Medium(c,\delta,\Delta) & \triangleq & \delta \leq T.\sigma_i - T.\sigma_{c(i)} < \Delta\\ Freshness(c,\delta,\Delta) & \triangleq & \delta \leq T.\sigma_{c(i)} - \hat{x}.\sigma_{c(i)} < \Delta\\ Fitness(c,\delta,\Delta) & \triangleq & \delta \leq Next(\hat{x}).\sigma_{c(i)} - T.\sigma_{c(i)} < \Delta\end{array}$$

When no lower (resp. upper bounds) is given, $0$ (resp. $+\infty$) is used.

These predicates have to be true at every state and every instant. The definition of an observation is done by giving which predicates must be satisfied. Other predicates can be proposed but we believe this set is sufficient to express the different behaviors that must be analyzed.

The first three predicates describe timed requirements on the system. $Lag$ is used to bound the lag introduced by the observation. The current value of the image was available on the source in one of the past instants. The bounds constrain how far in the past the image's current value is found on the source. They are the strongest needed to characterize all possible values of the image and so are defined using particular instants: the update instants of the image and the source. $Stability$ is used to get source values based on their duration. For example, we can eliminate transient values and keep sporadic ones, or the contrary. $Latency$ is a measure of the total duration between the current instant and the assignment of the image's current value on the source.

The last three predicates are used to define or restrict the behavior of the system due to its architecture. Predicate $Medium$ characterizes the medium implementing the observation, for example a communication bus or a processing unit. A lower bound is the shortest time needed to read the value of the source and assign it to the image. For example, it corresponds to the minimum communication time. An upper bound is the longest duration needed to create and send a message, for example, the maximum communication time plus the maximum time during which the source is unavailable.

When an observation denotes a computation, the bounds stand for the minimum execution time and the maximum execution time plus the blocking time (due to the scheduling policy for example). $Freshness$ and $Fitness$ are used to define intervals, relative to the update instants, where the value of the source is not available. The observation clock

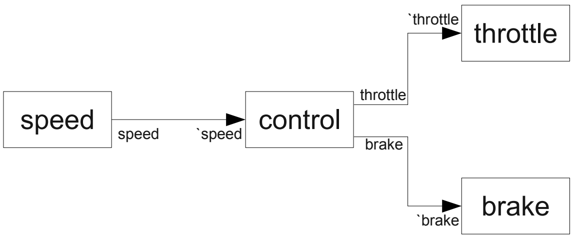


Fig. 3. Cruise Control System

is prevented from referring to one of these instants. An upper bound on $Freshness$ prevents values that are not fresh anymore to be read. On the contrary, a lower bound denotes an impossibility to access a value just after its assignment. $Fitness$ allows, or forbids, to read a value according to the time remaining until its removal; a lower bound prevents the value to be used just before a new update, for example when the computation of this new update has already started.

Note that, at the beginning of an execution, some predicates such as $Medium$ cannot be satisfied. In order to solve this problem, we define an observation where timed predicates do not have to be satisfied in initial states. The image values are replaced by a given default value. This extension is similar to the ”followed by” operator → in Lustre [9].

## V. A System Definition

### A. A Brief Description

We give here a system specification using previous properties. This system is a simplified car cruise control system used as an example.

The goal of such a system, when activated, is to control the throttle and the brakes in order to reach and keep a chosen speed. The system is based on different components (see Figure 3):

- a speed monitor which computes the current speed, based on a sensor counting wheel turns;
- the throttle actuator which controls the engine;
- the brakes which slow the car;
- the control system which handles the speed depending on the current and the chosen speed;
- a communication bus which links the devices and the control system.

The environment, the driver and the engine influence the speed of the car. Once the cruise control is activated and a speed is chosen, the control system chooses whether to accelerate by increasing the voltage of the throttle actuator or to decelerate by decreasing this voltage and by using brakes. The system has a maximum delay between each change to ensure a reactive behavior.

Each component uses and/or produces data. We use observations to specify the system and characterize correct executions. The notation is the one introduced when the timed observation and the variable behavior were defined.

### B. Relations Between Data

Firstly, we present some relations between the variables of the example. These relations, either computations or communications, are expressed as observations. The speed monitor computes the values of a variable $speed$ and these values are sent to the control system as a variable $`speed$. We express this as an observation $`speed \prec\!\!\cdot\; speed$.

The decisions of the control system are based on the current speed and more precisely on the value of $`speed$. Two functions are used to compute the values used as inputs by the brakes and the throttle actuator. Using the speed values, we compute the values of two variables: $throttle \prec\!\!\cdot\; control1(`speed)$ and $brake \prec\!\!\cdot\; control2(`speed)$.

Finally, the decisions $throttle$ and $brake$ are sent to dedicated devices into variables $`throttle$ and $`brake$, such that $`throttle \prec\!\!\cdot\; throttle$ and $`brake \prec\!\!\cdot\; brake$.

### C. Requirements and Properties

We state the requirements and known timed properties of the system and explain how to express them as characteristics of the system variables and observations. These characteristics are all given in Figure 4.

The speed is computed using the ratio between the number of wheel turns and the time. A minimum time is required to give a significant result. So the variable $speed$ must have a minimum time $\delta_1$ between each update. Due to scheduling constraints, computation of variables $throttle$ and $brake$ also have minimum times $\delta_2$ and $\delta_3$ between each update.

Each communication on the bus has a minimum communication time, regardless of the communicating protocol that is chosen. We introduce a lower bounded $Medium$ predicate on the observations denoting communication. Similarly there is also a lower bounded delay to represent the minimum computation time of the functions $control1$ and $control2$.

We expect each data to be used not too long after each update. More precisely we want each decision applied to the brake or to the throttle to be based on fresh values of the speed. Thus, we require the complete processing chain to be achieved in a short enough time.

A composition of observations is an observation, for example if $y \prec\!\!\cdot\; x$ and $z \prec\!\!\cdot\; f(y)$ then $z \prec\!\!\cdot\; f(x)$ [8]. We use this property to express the requirement on the complete processing chains between $`throttle$, $`brake$ and the variable $speed$. The definitions of these observations are compatible with the dependencies between the variables. The processing chains must satisfy upper bounded $Latency$ predicates on the observations. These are the only upper bounded characteristics given in the abstract system specification. Upper bounds on the $Medium$ and $Steadiness$ characteristics of the other observations and variables are implicitly imposed by the $Latency$ upper bound ($\Delta_7$).

### D. Case Study Analysis

The goal of the analysis is to prove that the specification is consistent and that there is at least one execution satisfying the requirements. In our example, the set of valid executions

`- variables behaviours:`

$$speed\ \{Steadiness(\delta_1, +\infty)\}$$
$$throttle\ \{Steadiness(\delta_2, +\infty)\}$$
$$brake\ \{Steadiness(\delta_3, +\infty)\}$$

`- communications:`

$$`speed \prec\!\cdot\ speed\ \{Medium(\delta_4, +\infty)\}$$
$$`throttle \prec\!\cdot\ throttle\ \{Medium(\delta_4, +\infty)\}$$
$$`brake \prec\!\cdot\ brake\ \{Medium(\delta_4, +\infty)\}$$

`- computations:`

$$throttle \prec\!\cdot\ control1(`speed)\ \{Medium(\delta_5, +\infty)\}$$
$$brake \prec\!\cdot\ control2(`speed)\ \{Medium(\delta_6, +\infty)\}$$

`- complete processing chains:`

$$`throttle \prec\!\cdot\ control1(speed)\ \{Latency(0, \Delta_7)\}$$
$$`brake \prec\!\cdot\ control2(speed)\ \{Latency(0, \Delta_7)\}$$

Fig. 4. System Specification

ensures the availability of timely sound values. From this set, we deduce the required update frequency of the *speed* variable. For example, we check the existence of a maximum time acceptable between each update. We analyze the admissible values of the $Medium$ to deduce the communication and computation times that are permitted. Finally, we determine the possible values of the observation clocks in the states corresponding to update instants of the image. These values give the instants at which the values of the source are caught and so, for example the instants when a message must be sent or a computation must start.

For all these properties, a choice must be done. For example, choosing a set of executions may alleviate the bounds on communication time but then reduce the instants where the message must be sent.

## VI. System Analysis

We give here properties of our framework based on observations in order to carry out an analysis. A system specified with observation relations must be analyzed to check the consistency of the specification, i.e. if there exists an execution satisfying the specification.

We discuss the analysis method in a discrete context. The semantics of the specification is restricted by discretizing time: i.e. the values taken by time $T$ are in $\mathbb{N}$. For discussion about the loss of information using discrete time instead of dense time and defending our choice, see [7] for example.

### A. Equivalence with a Finite System

Given a specification based on our framework, the value of $T$ is unbounded and we have no restriction on the values that can be taken by variables. Therefore the system defined by the specification is infinite. Nevertheless, we can build a finite system equivalent to the specification for the timed properties studied with this framework. This allows us to check the consistency of the specification in a finite time. Here are the main principles of this proof. The definition of a finite system bisimilar to the specified one is based on two equivalence relations.

Since the scope of this framework is to check the satisfaction of timed requirements, we focus on the auxiliary variables used to describe the timeline of each application variable. We define a system where only variables denoting instants are kept, i.e.. the variable describing the timelines and the observation clocks. The states and transitions of the system are defined by the values of these variables and the satisfaction of observations and variables properties. Allowed states and transitions do not depend on the values that can be taken by each variable but on the instants describing their timeline and on the observation clocks. Thus, when we build a system where only these instants are considered, we do not lose or add any characteristics about the timed behavior of the system. We define an equivalence where two states are equivalent if and only if the observation clocks and the variables denoting the update instants are equal. This equivalence is used to build a bisimilarity relation between the specified system and the one built upon only the instants.

The second reason preventing to consider a bounded number of states is the lack of bound on time. The values of the update instants and observation clocks are also unbounded. In order to reduce the possible values that can be taken by the system variables denoting instants, we define a system where all values of the instants are stored modulo the length of an analysis interval. We denote this number as $L$. $L$ must be carefully chosen, greater than the upper bounds on the variables $Steadiness$ and the observations $Latency$ characteristics and it has to be a multiple of the variable periods.

Such a number $L$ only exists if all variables and observations have upper bounded characteristics. When the source of an observation is bounded and so is the observation, such a bound is deduced for the image. Restricting the behavior by expecting variables to be frequently updated and the shift introduced by distribution to be bounded seems consistent for such real time systems.

In the system defined by the specification, transitions are based on differences between the instants characterizing the variable timelines. These differences cannot exceed the chosen length $L$. Thus, for each state, if the value of the time $T$ is known and if the values of the other variables are known modulo $L$, then for each variable there is only one possible real value that can be computed using the value of $T$. Consequently, considering the clock values modulo this length does not add or remove any behavior of the original system. We define an equivalence where two states are equivalent if the update instants and the observation clocks are equal modulo $L$. A system built by considering all values modulo $L$ is bisimilar with the original system using this equivalence.

Based on these two equivalences, we build a system by removing variables which do not denote update instants or observation clocks and by considering the values modulo $L$. This system is bisimilar to the specification and preserves the timed properties. Since all values are bounded by the length of the analysis interval and there is a bounded number of values, it defines a system with a bounded number of states. This result proves the decidability of the framework for the verification of safety properties that can be done using the finite system.

### B. Complexity

We have proved the existence of a finite system equivalent to our system. We give here the complexity of a process to effectively build this equivalent finite system. In order to build a transition from a state to a new state we build a set of inequalities deduced from the properties of the previous state and from the observation and variable properties. To solve this set of inequalities and so deduce the possible values of instant variables in the new state, we use difference bound matrices [10]. Considering a system where $n$ variables are studied, the size of each matrix is in $O(n^2)$ and thus the complexity for reducing it to its canonical form and so building the new state is in $O(n^3)$ [10]. The maximum number of states to build depends on all possible combinations of values taken by variables. Each timed variable can take values between $0$ and $L$ and the number of instant variables is a multiple of $n$ so we have $O(L^n)$ states. Lastly the complexity to build the system is in $O(n^3 * L^n)$ and considering the memory, we have to store $O(L^n)$ states and $O(L^{2n})$ transitions. Therefore this direct approach is technically feasible only with small enough systems.

### C. Other Approaches

Since our approach relies on the TLA+ formalism, we could have used the dedicated tool TLC, the TLA+ model checker. A logical definition of the observation requires the temporal existential quantifier $\boldsymbol{\exists}$, which is not implemented in TLC. Therefore a concrete definition of the observation based on an explicit observation clock has been used. It is only after we have reduced the system to a finite one that a model checker such as TLC could be used.

To be able to more precisely characterize executions satisfying the specification, we currently explore methods to build these executions more easily. A first proposal is to reduce the complexity of such a process by relying on proofs on system properties. The proof approach can easily be used only under certain conditions and in order to proceed to some system simplifications. For example, a periodic source induces properties for its image through an observation. Using these properties reduces the number of states we have to build by forecasting some impossible cases. Proving the full correctness of the system is possible but it is complex and it has not been automatized yet.

Another way is to use controller synthesis methods [11]. Properties of the observation can be expressed as safety properties using LTL and be derived as Bchi automata [12]. Two automata describe the behavior of the source and the image of an observation, exchanging values through a queue. Restrictions can be added to introduce the used implementation and its compatibility with executions defined by the specification. The complexity of controller synthesis methods has still to be explored.

## VII. Conclusion

We propose an approach focused on variables instead of tasks and processes, to model and analyze distributed real time systems. We specify an abstract model postponing task and communication scheduling. Based on the state transition system semantics extended by a timed referential, we express relations between variables and the timed properties of variables and communications. These properties are used to check the freshness of values, their stability, and the consistency of requirements. A possible analysis is to build a finite system bisimilar to the specified one. The results are used to help implementation choices.

Perspectives are to search other methods that decrease the complexity of the analysis of a specification and to use this approach with different examples to expand the number of available properties and increase expressivity. We also work on using analysis results to help generating an implementation satisfying the specification.

## References

[1] K. Tindell and J. Clark, "Holistic schedulability analysis for distributed hard real-time systems," *Microprocessing and Microprogramming—Euromicro Journal (Special Issue on Parallel Embedded Real-Time Systems)*, vol. 40, pp. 117–134, 1994.

[2] M. Xiong, R. Sivasankaran, J. A. Stankovic, K. Ramamritham, and D. Towsley, "Scheduling transactions with temporal constraints: exploiting data semantics," in *RTSS '96: Proc. of the 17th IEEE Real-Time Systems Symposium*, 1996, pp. 240–253.

[3] S. Anderson and J. K. Filipe, "Guaranteeing temporal validity with a real-time logic of knowledge," in *ICDCSW '03: Proc. of the 23rd Int'l Conf. on Distributed Computing Systems*. IEEE Computer Society, 2003, pp. 178–183.

[4] G. Roşu and S. Bensalem, "Allen Linear (Interval) Temporal Logic—Translation to LTL and Monitor Synthesis," in *International Conference on Computer-Aided Verification (CAV'06)*, ser. Lecture Notes in Computer Science, no. 4144. Springer Verlag, 2006, pp. 263–277.

[5] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.

[6] M. Abadi and L. Lamport, "An old-fashioned recipe for real time," *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 5, pp. 1543–1571, September 1994.

[7] L. E. A. and S.-V. A., "A framework for comparing models of computation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 12, pp. 1217–1229, Dec. 1998.

[8] M. Charpentier, M. Filali, P. Mauran, G. Padiou, and P. Quinnec, "The observation : an abstract communication mechanism," *Parallel Processing Letters*, vol. 9, no. 3, pp. 437–450, 1999.

[9] N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud, "The synchronous data-flow programming language LUSTRE," *Proceedings of the IEEE*, vol. 79, no. 9, pp. 1305–1320, September 1991.

[10] J. Bengtsson and W. Yi, "Timed automata: Semantics, algorithms and tools," in *Lecture Notes on Concurrency and Petri Nets*, ser. Lecture Notes in Computer Science vol 3098, W. Reisig and G. Rozenberg, Eds. Springer-Verlag, 2004.

[11] E. Asarin, O. Maler, and A. Pnueli, "Symbolic controller synthesis for discrete and timed systems," in *Hybrid Systems II*. London, UK: Springer-Verlag, 1995, pp. 1–20.

[12] M. Y. Vardi, "An automata-theoretic approach to linear temporal logic," in *Logics for Concurrency: Structure versus Automata, volume 1043 of Lecture Notes in Computer Science*. Springer-Verlag, 1996, pp. 238–266.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# CompactRIO Embedded System in Power Quality Analysis

Petr Bilik
Dpt. of Electrical Measurement
VSB-Technical University
Ostrava, Czech Republic
Email: petr.bilik@vsb.cz

Ludvik Koval
Dpt. of Electrical Measurement
VSB-Technical University Ostrava,
Czech Republic
Email: ludvik.koval@vsb.cz

Jiri Hajduk
student
VSB-Technical University
Ostrava, Czech Republic
Email: jhajduk@volny.cz

***Abstract*—Electrical measurement department of VSB-Technical University has been involved for more than 14 years in research and development of Power Quality Analyzer built on Virtual Instrumentation Technology. PC-based power quality analyzer with National Instruments data acquisition board was designed and developed in this time frame. National Instruments LabVIEW is used as the development environment for all parts of power quality analyzer software running under MS Windows OS. Proved PC-based firmware was ported to new hardware platform for virtual instrumentation – National Instruments CompactRIO at the end of 2007. Platform change from PC to CompactRIO is not just code recompilation, but it brings up many needs for specific software redesigns. Paper describes how the monolithic executable for PC-based instruments was divided into three software layers to be ported on CompactRIO platform. The code for different parts of CompactRIO instrument is developed in a unified development environment no matter if the code is intended for FPGA, real-time processor or PC running Windows OS.**

***Keywords*—CompactRIO, FPGA, VxWorks, LabVIEW, Power Quality**

## I. Introduction

THE MEMBERS of the Electrical Measurements Department have been involved into the research and development in the area of electrical power quality for more than fourteen years. During this period the PC-based power quality analyzer has been developed.

The basis of the solution is the software application designed and developed in the graphical development environment LabVIEW and running on a PC. The core of the application is the data acquisition process with adaptive sampling frequency. The sampling frequency and the length of the time window are chosen to follow the requirements of elemental standards for power quality area: IEC 61000-4-30 and IEC 61000-4-7. The software provides so called „gap free“ measurement by continuous data acquisition. The algorithm in conjunction with the 16-bit plug-in data acquisition board allows change of the sampling frequency on the fly. This eliminates measurement errors in frequency domain caused by the leakage that normally appears when the frequency of the measured signal is changing during the measurement period.

Under this data acquisition core the software modules provide particular instrument functionality from the user point of view, such as FFT analyzer, power and energy analyzer, voltage quality analyzer etc. The power quality analyzer allows analyzing up to 4 voltages and 4 currents.

The concept of PC-based instrumentation has been used by measurement instrument manufacturers for long time period; it has many advantages, but it is difficult to design such instruments with small size and low power consumption. A new possibility in this area is brought by the CompactRIO hardware platform described in this paper.

The goal of this paper is the description of software and instrument firmware developed by the authors for this particular problem.

## II. Virtual instrumentation

The instrumentation has often evolved into computer-based measurement systems. These systems are built with measurement hardware consisting of a digitizing element whose performance is characterized by its resolution and sample rate. The performance of this hardware, including bandwidth, accuracy and resolution, has increased dramatically in the past 10 years to the extent that the performance of the computer-based digitizers is comparable to traditional instruments. Virtual instruments are defined by the user while traditional instruments have fixed manufacturer-defined functionality, see Fig.1

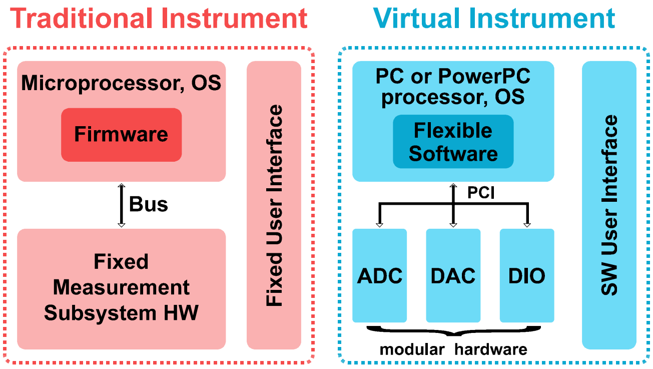


Fig. 1. Traditional instruments (left) and software based virtual instruments (right) share the same architectural components, but radically different philosophies

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## III. PC based Power Quality Analyzer

The basic components of PC based power quality analyzer are common, no matter whether it is a desktop PC, stationary instrument or hand-held instrument.

The PC is equipped with the plug-in data acquisition board from National Instruments. Currently the M-series PCI and USB boards are supported. M-series boards provide 16 analogue inputs and the aggregated sampling rate is at least 250 kS/s. The data acquisition process of instrument SW uses sampling frequency 9.6 kS/s per channel.

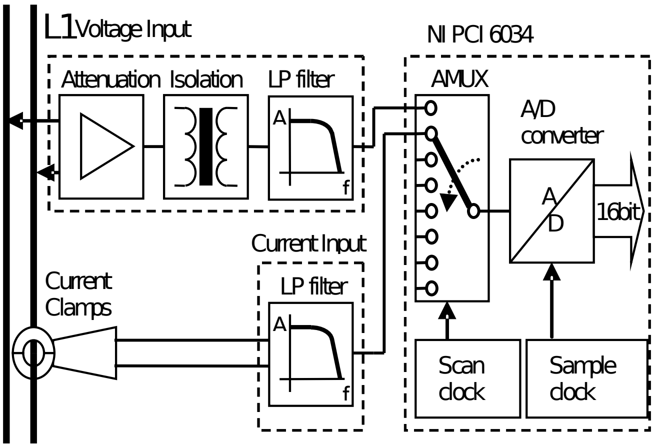


Fig. 2. Power Quality Analyzer input circuit diagram.

Modules for signal conditioning of voltage and current signals provide attenuation / amplification, isolation and anti-aliasing filtering. Modules for signal conditioning are programmable via digital lines or serial interface. This allows inputs range selection and anti-aliasing cut-off frequency set-up. The block diagram of signal conditioning modules connected to analog to digital converter is shown on Fig. 2.

The operating system used for the PC based instrument is Windows XP or Windows XP Embedded. Power quality analyzer firmware fully written in LabVIEW is running in the operating system as EXE application. In case if instrument is equipped with a touch-in display, the user can operate the instrument without keyboard or mouse, because complete firmware is ready to use soft-keyboards.

## IV. Signal Sampling According To IEC61000-4-30

According to IEC61000-4-30 the basic measurement time interval for voltage quality parameters (supply voltage, harmonics, interharmonics and unbalance) shall be a 10-cycle time interval for 50 Hz power system or 12-cycle time interval for 60 Hz power system. It is not a fixed interval in time, but measurement interval varies in time as the fundamental frequency of power network changes. The adaptive sampling frequency solves this requirement and avoids not wanted phenomena in frequency domain like leakage and scalloping loss.

The easiest way to achieve adaptive sampling frequency is PLL (Phase Lock Loop). Data acquisition boards are not equipped with PLL hardware and thus software version of PLL principle was developed, see Fig.3. The most important on the “software PLL” is a very precise frequency measurement algorithm from 12 periods of signal.

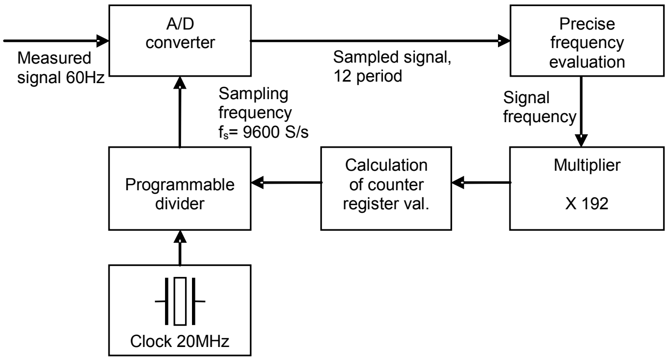

Measured signal 60Hz
A/D converter
Sampled signal, 12 period
Precise frequency evaluation
Sampling frequency $f_s$= 9600 S/s
Signal frequency
Programmable divider
Calculation of counter register val.
Multiplier
X 192
Clock 20MHz

Fig. 3. Software implementation of PLL

The method using envelope curve of the spectral lines in frequency domain was used. If rectangular window is used in time domain then the envelope curve in frequency domain is a function *sin(x)/x.* The precise frequency can be calculated from amplitudes of two nearest spectral line from signal frequency, even if the frequency is not integer multiply of the step in frequency domain.

## V. Analyzed quantities and Measurement uncertainty achievment

According to IEC61000-4-30 Power Quality Analyzer should analyze and evaluate these quantities: power frequency, magnitude of the supply voltage, flicker, harmonics and interharmonics, supply voltage unbalance, rapid voltage changes and voltage dips, swells and interruptions. It is suggested that currents will be also monitored and analyzed. For any mentioned quantity the measurement uncertainty is specified. The requested uncertainty on voltage and current magnitude should be better than 0.1% for the complete instrument including sensors (e.g. current clamps).

To understand the type and magnitude of errors, FLUKE 6100A Electrical Power Standard programmable calibrator has been used. Numerous errors were discovered: frequency amplitude errors, frequency phase errors, amplitude and phase error (depending on the magnitude of signal within the measurement range). After thorough analysis of the discovered errors a solution was designed to accomplish the error compensation. Some errors can be corrected in time domain; some errors must be corrected in frequency domain. An appropriate structure of calibration tables was designed and implemented as automated calibration software.

## VI. CompactRIO Hardware Description

CompactRIO does not replace PC-based instruments for virtual instrumentation, but it is collateral hardware platform with smart structure.

CompactRIO combines an embedded floating point processor (PowerPC) with real-time operating system VxWorks, a high-performance FPGA and hot-swappable I/O modules. Each I/O module is connected directly to the

FPGA, providing low-level customization of timing and I/O signal processing. The FPGA is connected to the embedded real-time processor via a high-speed PCI bus, see Fig.4. This represents architecture with open access to low-level hardware resources. Both PowerPC processor and FPGA are programmed in graphical programming language LabVIEW. LabVIEW contains built-in data transfer mechanisms to pass data from the I/O modules to the FPGA and also from the FPGA to the embedded processor for real-time analysis, postprocessing, data logging, or communication to a networked host computer.

CompactRIO in comparison with PC-based instruments brings compact solution, small size 88 x 180 x 90mm (HxWxD), wide operating temperature -40°C to +70°C and very low power consumption (approximately 8W).

## VII. CompactRIO Based Quality Analyzer

After long time development and improvements of PC-based Power Quality Analyzer it was decided to port the proved PC code to CompactRIO. The original code of PC-based instrument was divided into three layers:

- FPGA (data acquisition process, SW PLL)
- real-time processor, (data analysis functions)
- host PC (user interfaces)

All three described parts were implemented in the same development environment: National Instruments LabVIEW. Very powerful part of CompactRIO is FPGA. It can solve functionality between I/O modules without interaction with real-time processor. Fixed point arithmetic library and even FFT routines for FPGA are available in LabVIEW. The code on FPGA is extremely fast in comparison with embedded processor.

It was decided to move adaptive sampling frequency algorithm directly on FPGA. All blocks on Fig.3 are implemented on FPGA. FPGA ensures sampling of 3 voltage signals and 4 current signals. Other parallel task running on FPGA is mechanism of data transfer to real-time processor, digital input acquisition and transfer, GSM modem data transfer. LabVIEW code implementation for FPGA is not fast and easy, because compilation of this part of code takes several tens of minutes. As an optional way of software PLL was developed resampling algorithm. Resampling algorithm was implemented on FPGA and provides signal samples with different sampling frequency than AD converter. This algorithm is still under testing process.

PowerPC processor runs VxWorks operating system and LabVIEW application with several parallel tasks. Application receives data samples from FPGA and calculates from them many parameters including: RMS values, frequency, harmonic spectra, total harmonic distortion, flicker, three-phase system unbalance, active power, reactive power, energies and many other quantities. Calculated quantities are aggregated in time and some data are statistically evaluated before storing. Data are stored periodically with defined time interval and event-based data are stored just in time when event appears. Data are stored to local solid state disc. Real-time processor ensures also TCP communication with host PC. LabVIEW code implementation for PowerPC is fast because LabVIEW provide on the fly compilation of code fragments.

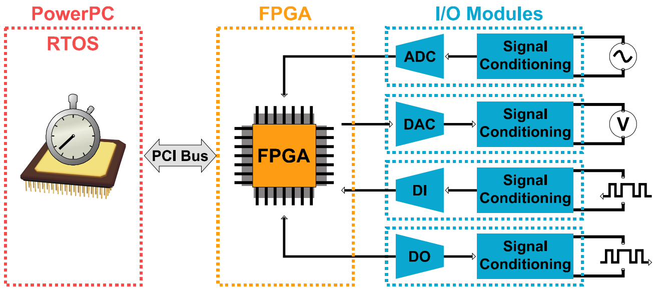


Fig. 4. Compact-RIO internal structure

Host PC applications, also developed in LabVIEW, serve as graphical user interface for CompactRIO instrument. CompactRIO and host PC are connected each other by Ethernet and they use TCP protocol to communicate.

## VIII. Power Quality Analyzer Software Suite

The project name of complete software suite for power quality analyzer is ENA. Directly on CompactRIO runs ENA-Node firmware. ENA-Node runs on FPGA and on PowerPC embedded in CompactRIO.

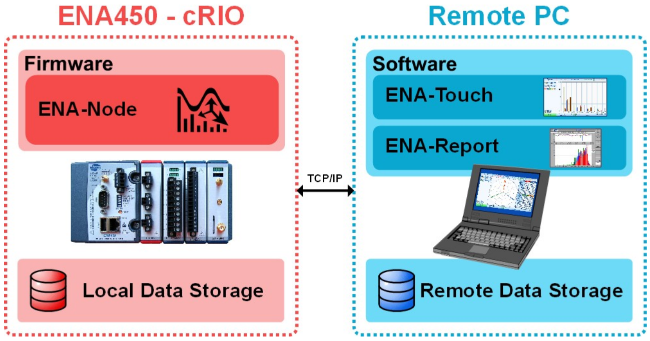


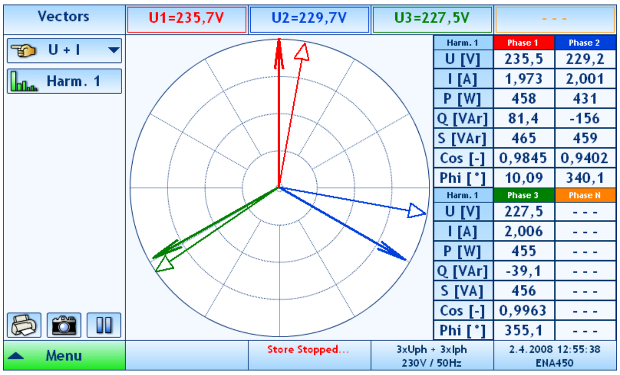
Fig. 5. ENA Power Quality Analyzer Software Suite

The user interface ENA-Touch runs on host PC connected to ENA450 and allows instrument set-up and on-line data visualization, see Fig.5.

The main concern of ENA-Touch development cycle was user friendliness, ease of operation and compliance with international standards. ENA-Touch can control ENA-Node remotely by using TCP protocol over Ethernet. New original approach to instrument set-up and to display measured quantities allows easy and well arranged configuration and measured data presentation. In ENA-Touch exist numerous ways how to present data: tables, time and frequency domain graphs, scope, vectorscope (see Fig.6) and statistical results for power quality.

## IX. Conclusion

CompactRIO is a combination of powerful hardware and software running on real-time processor and FPGA in parallel. CompactRIO brings even better flexibility for virtual instrumentation than PC platform. The same graphical development environment LabVIEW used for three levels of hardware FPGA, real-time processor and PC dramatically simplifies the SW maintenance and further



| Harm. 1 | Phase 1 | Phase 2 |
|---|---|---|
| U [V] | 235,5 | 229,2 |
| I [A] | 1,973 | 2,001 |
| P [W] | 458 | 431 |
| Q [VAr] | 81,4 | -156 |
| S [VAr] | 465 | 459 |
| Cos [-] | 0,9845 | 0,9402 |
| Phi [°] | 10,09 | 340,1 |

| Harm. 1 | Phase 3 | Phase N |
|---|---|---|
| U [V] | 227,5 | - - - |
| I [A] | 2,006 | - - - |
| P [W] | 455 | - - - |
| Q [VAr] | -39,1 | - - - |
| S [VA] | 456 | - - - |
| Cos [-] | 0,9963 | - - - |
| Phi [°] | 355,1 | - - - |

Fig. 6. ENA-Touch user interface example - Vectorscope

development. The development process for CompactRIO is not as fast as on PC platform, but there is no need to know VHDL language to program FPGA, no need to be deeply familiar with real-time operating system VxWorks. The main advantage of described CompactRIO-based solution is its flexibility and scalability. Big support of advanced math functions for FPGA becomes available; this will allow moving many calculation routines from real-time processor to FPGA.

Porting of PC-based instrument code to CompactRIO hardware platform was not a trivial task. There are many differences between PC and CompactRIO approaches to the same problem. Even an author with more than 12 years of experience and deep knowledge of LabVIEW needs several months to have fully functional and tested solution ready to be a product. There is a long way from simple CompactRIO demo showed on trade shows to functional and reliable complex instrument. The goal of the team was to test if CompactRIO hardware together with LabVIEW graphical development environment is a usable platform for future implementations of such complex measurement instruments like power quality analyzer is. The project confirms that a suite CompactRIO and LabVIEW is a reliable platform for virtual instrumentation of this sort.

## References

[1] P. Bilik, J. Zidek, “Disturbance Recorder as a Part of PC Based Power Quality Analyzer,” in *Conference proceeding of IVth International Scientific Symposium EE 2007*, Technical University of Kosice, 2007. ISBN 978-80-8073-844-0.

[2] P. Bilik, J. Zidek, D. Kaminsky, J. Hula, M. Malohlava, M. Rumpel, “EPQA ENA100—Handheld Power Quality Analyzer Powered by LabVIEW,” in *Conference proceedings of NI Week 2007*, Austin: National Instruments, 2007.

[3] P. Bilik, J. Hula, L. Koval, “Modular System For Distributed Power Quality Monitoring,” in *Conference proceeding of Power Quality and Utilization EPQU 2007*, Electrical Engineering Department TU of Catalonia, 2007. ISBN 978-84-690-9441-9.

[4] EN 50160: Voltage characteristics of Electricity supplied by Public Distribution Systems

[5] IEC 61000-4-7 Amend.1 to Ed.2: Electromagnetic compatibility (EMC): Testing and measurement techniques - General guide on harmonics and interharmonics measurements and instrumentation, for power supply systems and equipment connected thereto

[6] IEC 61000-4-15 Electromagnetic compatibility (EMC): Testing and measurement techniques—Flickermeter—Functional design specifications

[7] IEC 61000-4-30 Electromagnetic compatibility (EMC): Testing and measurement techniques – Power quality measurement methods

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Real-time Task Reconfiguration Support Applied to an UAV-based Surveillance System

Alécio Pedro Delazari Binotto
Fraunhofer IGD / TU Darmstadt – Germany
PPGC UFRGS – Brazil
Email: alecio.binotto@igd.fraunhofer.de

Edison Pignaton de Freitas
IDE – Halmstad University – Sweden
PPGC UFRGS – Brazil
Email: edison.pignaton@hh.se

Carlos Eduardo Pereira
PPGC UFRGS – Brazil
Email: cpereira@ece.ufrgs.br

André Stork
Fraunhofer IGD / TU Darmstadt – Germany
Email: andre .stork@igd.fraunhofer.de

Tony Larsson
IDE – Halmstad University – Sweden
Email: tony.larsson@hh.se

*Abstract*—**Modern surveillance systems, such as those based on the use of Unmanned Aerial Vehicles, require powerful high-performance platforms to deal with many different algorithms that make use of massive calculations. At the same time, low-cost and high-performance specific hardware (e.g., GPU, PPU) are rising and the CPUs turned to multiple cores, characterizing together an interesting and powerful heterogeneous execution platform. Therefore, reconfigurable computing is a potential paradigm for those scenarios as it can provide flexibility to explore the computational resources on heterogeneous cluster attached to a high-performance computer system platform. As the first step towards a run-time reconfigurable workload balancing framework targeting that kind of platform, application time requirements and its crosscutting behavior play an important role for task allocation decisions. This paper presents a strategy to reallocate specific tasks in a surveillance system composed by a fleet of Unmanned Aerial Vehicles using aspect-oriented paradigms in order to address non-functional application timing constraints in the design phase. An aspect support from a framework called DERAF is used to support reconfiguration requirements and provide the resource information needed by the reconfigurable load-balancing strategy. Finally, for the case study, a special attention on Radar Image Processing will be given.**

## I. Introduction

In addition to timing constraints, several modern applications usually require high performance platforms to deal with different algorithms and massive calculations, varying from monitoring and processing of different data acquired by sensors to cryptography and large image data visualizations and processing. The development of low-cost powerful and application specific hardware (for example, the GPU —Graphics Processing Unit, the Cell processor, PPU—Physics Processing Unit, DSP—Digital Signal Processor, PCICC—PCI Cryptographic Co-processor, FPGA—Field Programmable Gate Array, among others) offer several execution alternatives aiming better performance, programmability and control. The resulting execution platform heterogeneity is intensified with multi-core CPUs, causing problems to achieve simple programming and efficient resource utilization.

Following that direction, low-cost inter-chip hybrid hardware architectures are becoming very attractive to compose adaptable execution platforms and, at the same time, software applications must benefit from that performance powerfulness. This leads to the creation of new methods and strategies to distribute the application's workload (tasks, algorithms, full applications) to execute in the specific processing units in order to better meet application requirements, such as performance and timeliness, without loosing flexibility. In this manner, reconfigurable load-balancing computing is a potential paradigm for those scenarios as it can provide flexibility, improve efficiency, and offer simplicity to high performance heterogeneous processor cluster and multi-core architectures. Figure 1 shows such a theoretical scenario.

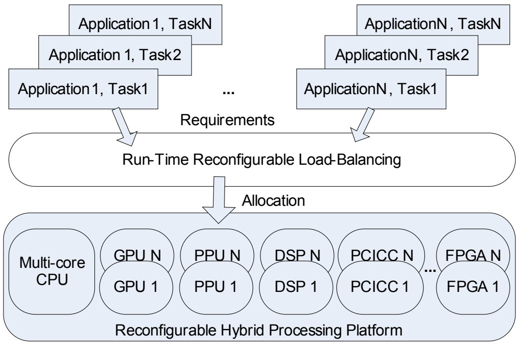


Fig. 1. System overview

The goal is to design proper methodologies, strategies, and support for management of a dynamic load-balancing by means of real-time reconfiguration of involved tasks. To reach this objective, thereby, the main step is to create a self reconfiguration framework that can be used by applications composed of different kinds of algorithms (graphics, massive mathematical calculations, sensor data processing, artificial intelligence, cryptography, etc) which runs on a single personal computer and needs to execute under time constraints and a minimal quality (performance). In addition, during execution time, the framework is intended to keep monitoring the tasks and provides online information in order to a possible new allocation balance, i.e., a reconfiguration of tasks is done if this procedure can promote a better performance for the overall current scenario.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

In this paper, the focus is on the very first step in the design method framework: application requirements handling (reconfiguration) in a high-level design. The approach is to extract the requirements to find concurrency in order to base the load-balancing framework towards task parallelization and reallocation. For its accomplishment, the crosscutting concerns related to the real-time non-functional requirements are taken into account. The handling of these concerns by specific design elements called “aspects” (from aspect-oriented paradigm [1] plays an important role to improve understandability and maintainability during task (re)configuration. Based on the support offered by the aspects to monitor and control the above mentioned requirements, a strategy to assign tasks dynamically to units of execution is presented, being these tasks subject to an on-line reconfiguration.

The paper is organized as follows. We start in Section II with a previous work based on aspects and a description of the created ones for requirements identification, modeled using UML notation. Section III follows by a reconfigurable workload strategy implemented by the aspects. Composing these two concepts, Section IV outlines Unmanned Aerial Vehicle Surveillance system as application, focusing on radar image processing tasks. Finalizing, related works, conclusions and future works are presented.

## II. Aspect-oriented Handling of Real-time Concerns

In order to provide a reconfigurable solution in runtime with the goal to meet timing requirements, several mechanisms to control and monitor timing parameters must be inserted in the system. Moreover, the mechanisms related to the migration of tasks among processing units, which implements the system reconfiguration itself, also affect several elements in the system in a non-uniform way. Besides, all these mechanisms and controls are not the main goal or functionality of any system, but they must be present in order to achieve reconfiguration. These facts are clear characterized as non-functional crosscutting concerns, which can be successfully addressed by an aspect-orientation [1].

The non-functional requirements handling concerns hinder the system maintainability, reuse, and evolution in current approaches used in task reconfiguration such as those that use pure object-orientation. It occurs because the handling elements (such as timing requirements probes, serialization mechanisms, task migration mechanisms, among others) are not modularized in a single or few system elements, but spread over the system. Any change in one of these elements requires changes in different parts of the system, what besides to be a tedious and error-prone task, do not scale in the development of large and complex applications. The observation of these drawbacks motivates the use an aspect-oriented approach that makes possible to address such concerns in a modularized way. It separates the handling of the non-functional concerns in specific elements, increasing the system modularity, diminishing the coupling among elements, and though affecting positively the system maintainability, reuse and evolution. Moreover, the advantages of the aspect-oriented approach became clearer when applied to the aforementioned task allocation strategies using heterogeneous platforms due to the need of profiling each task in a different hardware, affecting several elements of the application. The use of aspects to address this concern represents an improvement as it helps to cope with the complexity in managing this handling that is spread allover the system.

In the next subsection, the aspects used to address timing related concerns will be presented, which come from the DERAF framework [2] that provides a high-level aspect library to handle Distributed Real-time Embedded (DRE) systems non-functional requirements.

### *A. Time-based Aspects*

In order to support Timing and Precision requirements, the proposal is to use some aspects from the DERAF framework. The packages designed to these types of requirements are presented in Figure 2 and a short description of each one is provided in the following paragraphs. Interested readers are referred to [2] and [3] for more details.

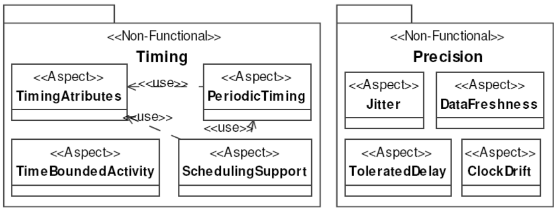


Fig. 2. Timing and Precision packages from DERAF

TimingAttributes: adds timing attributes to active objects (e.g., deadline, priority, start/end time, among others), and also the corresponding initialization of these attributes.

PeriodicTiming: adds a periodic activation mechanism to active objects. This improvement requires the addition of an attribute representing the activation period and a way to control the execution frequency according to this period.

SchedulingSupport: inserts a scheduling mechanism to control the execution of active objects. Additionally, this aspect handles the inclusion of active objects into the scheduling list, as well as the execution of the feasibility test to verify if the scheduling list is schedulable.

TimeBoundedActivity: temporally limits the execution of an activity by adding a mechanism or changing a parameter that can restrict the maximum execution time for an activity (e.g., limit the time which a shared resource can be locked).

Jitter: measures the start/end of an activity, calculates the variation of these metrics and whether the tolerated variance was overran.

ToleratedDelay: temporally limits the beginning of an activity execution (e.g., limits the time which an active task can wait to acquire a lock on a shared resource).

DataFreshness: associates timestamps to data, verifying their validity before using them [4].

ClockDrift: measure the time at which an activity starts and compares it with the expected beginning of this activity; it also checks if the accumulated difference exceeds the maximum tolerated clock drift.

Additional aspects were developed to compose the information provided by the DERAF aspects to inject the reconfiguration strategy. These aspects as described in the following.

### *B. Aspects for Supporting Reconfiguration*

As mentioned before, the task migration support characterizes a non-functional crosscutting concern that affects different parts of the system in different ways. On this way, we propose the use of aspects to address this concern. The new aspects proposed in this work are: **TimingVerifier**, **TaskAllocationSolver**.

Both use the time parameters inserted by the aspects of the Timing package, described previously, and the services provided by the aspects from the Precision package. An additional package from DERAF is also used in order to make the reconfiguration decisions take place; it is the TaskAllocation package, which is presented in more details latter on in this section. Figure 3 depicts the schema.

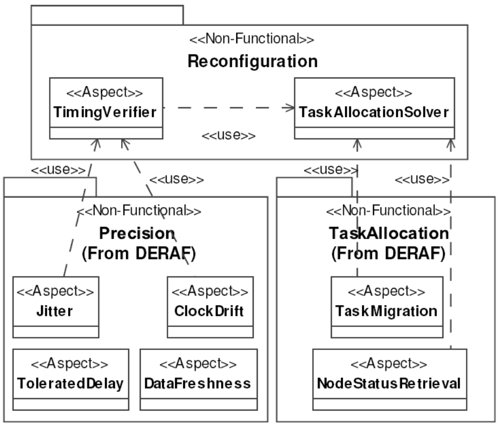


Fig. 3. Aspects for Reconfiguration

The **TimingVerifier** is responsible for checking if the processing units are being able to accomplish with the timing requirements specified by the TimingAttributes, PeriodicTiming, ToleratedDelay and TimeBoundedActivity aspects. In order to do this, it uses the services from the aspects Jitter and ClockDrift.

A mechanism to control the meeting of timing attributes is inserted in the beginning and end of each task. This mechanism consists of measuring these times, and comparing them with the requirement specified by the correspondent attribute. As an example, the accomplishment of a specified deadline can be checked by measuring the time in which the task actually ended its computation and comparing it with the time in which it was supposed to finish. It uses the service of the Jitter aspects to gather and analyze information about the jitter related to the corresponding requirement. Taking the deadline again as an example, it measures if a non-accomplishment of a deadline is constant or if the measure varies in different executions or in changing the platform scenario. It can be used, for instance, as base information to know if the interaction with other tasks is being responsible for the variance.

The ClockDrift aspect is used by the TimingVerifier to gather information about the synchronization among the different Processing Units (PUs). It is useful to calculate the cost, in terms of a task migration time. In order to illustrate the idea, consider a task that was migrated from a PU "A" to a PU "B". The difference in the clock drift between them can result in a waiting time for the result from the PU "B" that does not worth if compared with leaving the task running in the PU "A".

**TaskAllocationSolver** is the second key aspect. It is responsible for deciding whether a task will be migrated or not and to which of the available PUs. It also has to check possible overload of the PU candidate destinations in order to decide whether it is worthwhile to perform the migration. The TaskAllocationSolver uses the measurements available due to the work done by the TimingVerifier.

Based on these data, the reasoning about the feasibility or not of a task reconfiguration is done. The explanation of this reasoning is provided in the next section, where the reconfiguration strategy is presented.

The reconfiguration itself and the retrieval of PUs (nodes) status are done by two other aspects from DERAF, the NodeStatusRetrieval and the TaskMigration. This way, the reasoning and the performance of the reconfiguration are decoupled, allowing that changes performed in one aspect do not affect the other. A brief summary of the NodeStatusRetrieval and TaskMigration aspects is provided in the following.

TaskMigration: provides a mechanism to migrate active objects (tasks) from one PU node to another PU node. It is used by the aspects that control embedded concerns and, in the present work, by the allocation solver aspect TaskAllocationSolver.

NodeStatusRetrieval: inserts a mechanism to retrieve information about processing load, send/receive message rate, and/or the PU availability (i.e., the "I'm alive" message). Before/after every execution start/end of an active task, the processing load is calculated. Before/after every sent/received message, the message rate is computed. Additionally, the PU availability message is sent at every "n" message or periodically with an interval of "n" time units.

## III. Dynamic Reconfiguration Strategy

A task-based decomposition approach is used, in which each task is an independent algorithm. The tasks are grouped according derivation of the same high-level, simplifying the managing of possible dependencies. Besides, it is coherent to assume that a group of tasks will have the same characteristics or, in other words, the same designed features and hence would be desirable to execute in the same PU. However, this can lead to a non-optimal execution, as it will be discussed in the next sub-sections.

### *C. First Assignment of Tasks*

The strategy is to combine a costly method for assignment problems with a real-time measurement procedure based on the mentioned aspects. For the first assignment of tasks, we do not use the modeled aspects, since we do not have real time measurements of tasks. Therefore, we decided to offer two possibilities: assign all the tasks in the first time step to the CPU and then perform the dynamic reconfiguration; or model the first assignment as a common assignment problem using Integer Linear Programming (ILP), like the generic approach used by the authors of [5]. In this way, a set of tasks can be represented as $T = \{t_{i,j}\}$, where every task $i$ has an implementation based on the specific hardware API (considering each supported hardware) and an execution cost estimation on the PU $j$. The execution cost $c_{i,j}$ is simply estimated, in this paper, based on a correlation between the input data type used by the task (mapped within a weight scale) and the number of sub-cores presented in the corresponding processing unit. A flag is assigned for tasks that only execute in determined hardware. Note that these costs could not reproduce the reality with fidelity and are just a way to determine a "first guess" to the system.

The total execution time of the application is minimized by finding a schedule solution by means of its tasks execution times. The constraints for the presented model will be the maximum workload of each processing unit ($U_{max}$), which are represented by

$$U_j = \sum_{i=1}^{n} x_{i,j} c_{i,j} \leq U_{j_{\max}} \quad (1).$$

The objective function that minimizes the total application execution time is, then, defined as

$$\min \left\{ \sum_{j=1}^{m} \sum_{i=1}^{n} x_{i,j} c_{i,j} \right\} \quad (2),$$

being the variables $x_{i,j}$ the solution for the modeled ILP.

The ILP problem is considered of complexity NP-hard and it is costly to calculate every period of time to estimate the current optimal assignment. Due to that reason, some approaches concentrate on heuristic-based methods to estimate the best assignment, as the above mentioned work of [5]. Nevertheless, this direction neither considers real execution times nor could represent the best assignment.

In counter part, the approach presented in this paper allows taking into account real execution measurements extracted from the processing units (using aspects) and works with a dynamic reconfiguration module that deals with real execution variables and interferences, leading to a possible better task assignment.

### *D. Real-time Reconfiguration*

After the first tasks assignment, the dynamic reconfiguration strategy enters on scene in application run-time. At this point, we consider the information provided by the created aspects and the first assignment. The assumption is: based on involved migration costs and possible interferences of new loaded tasks, one task can be reconfigured to run in other PU just if the new PU already finished the processing of its tasks (is idle) and the estimated time to execute the task in the new PU will be less than the time to execute in the actual PU, i.e., just if there is a gain. In a simple equation, this relationship can be modeled in terms of the costs:

$$T_{reconfigPUnew} < T_{remainingPUold} - T_{estimatedPUnew} - T_{overhead} \quad (3),$$

where the remaining time ($T_{remainingPUold}$) and the estimated time ($T_{estimatedPUnew}$) are calculated, respectively, for the current PU and for the new candidate PU based on previous measurements or on the first assignment (in the case of first reconfiguration invocation); and an overhead ($T_{overhead}$) that explicit the execution time of the reconfiguration itself. The relationship between $T_{remainingPUold}$ and $T_{estimatedPUnew}$ is considered the partial gain.

The information needed to calculate the reconfiguration will be provided by the `TimingVerifier` aspect and can be modeled without such mathematical formality as:

$$T_{reconfigPUnew} = T_{setupReconfigPUnew} + T_{temporaryStorage} + T_{transferRate} + T_{executionPUnew} + L \quad (4),$$

where $T_{setupReconfigPUnew}$ represents the time for setting up a new configuration on the new PU; $T_{temporaryStorage}$ contributes with the time spent to save temporal data if needed (considering shared and global memory access parameters); $T_{transferRate}$ measures the cost for sending/receiving data from/to the CPU to/from the new PU, which can be a bottleneck on the whole calculation; $T_{executionPUnew}$ symbolizes the measured or estimated cost of the task processed in the new PU; and finally $L$ denotes a constant to represent possible system latency.

During the application execution, the load-balancing module will keep storing the execution times for each type of the tasks with the information gathered by the `TimingVerifier` aspect together with the data provided by the `NodeStatusRetrieval` aspect. These preliminary stored times will be useful as one of the basis to the reconfiguration decision done by the `TaskAllocationSolver`.

An overview about the created strategy is presented on Tables 1 and 2. The behaviors that compose this strategy are inserted in the core of the system by the above mentioned aspects and those used by them, as presented in section II.B.

Table I.
Task Reallocation Algorithm

```
1:Acquire Timing data about tasks execution;
2:Acquire data about PUs processing load;
3:Calculate the Execution Priorities for all tasks, including new
    loaded ones, based on steps 1 and 2;
4:Calculate the New Load-Balance, taking into account the new
    priorities calculated previously;
5:Compare the New Load-Balance in accordance with equation (3)
    and, consequentially, equation (4);
6:Perform the Reconfiguration Decision;
7:Migrate tasks to perform the reconfiguration when applicable.
```

Table II.
Load-balancing Module Algorithm

```
1:Detect the Available PUs;
2:Calculate the First Assignment using equation (1) and (2) or assign
    all tasks to CPU;
3:For each n time-step do:
4:  Execute Task Reallocation Algorithm;
5:  Store measured times;
6:End For.
```

## IV. UAV-based Area Surveillance System

The presented ideas are illustrated by a case study that consists of a fleet of Unmanned Aerial Vehicles (UAVs) used in area surveillance missions. Such UAVs can be equipped with different kinds of sensors that can be applied, depending on the weather conditions, time of the day and goals of the surveillance mission [6]. We consider a fleet of UAVs that might accomplish missions during all the day and under all weather conditions. The UAVs also must be able to provide different levels of information precision and detail, depending on the required data.

To start, the UAVs receive a mission to survey a certain area and provide required data according to the mission directives. Their movements are coordinated with the other UAVs in the fleet in order to avoid collisions and also provide optimum coverage of the target area. Details about the used coordination algorithm are out of the scope of this paper.

Each UAV is composed of six subsystems that make it able to accomplish its missions and coordinate with the other UAVs. These subsystems are: Collision Avoidance, Movement Control, Communication, Navigation, Image Processing, and Mission Management.

In order to run the tasks described above and meet the high-lighted requirements and constrains modeled on the previous sections, we consider UAVs equipped with the following sensors: Visible Light Camera (VLC); Synthetic Aperture Radar (SAR) and Infra-Red Camera (IRC). In order to support the movement control, communication devices and embedded sensors, each UAV will be equipped with a hybrid processing unit target platform which is used accordingly to the specific needs during the accomplishment of a certain mission, detailed in the following.

### *E. UAV Subsystems*

As mentioned above, each UAV has six subsystems. Each subsystem has a number of tasks that perform specialized activities related to a specific functionality. An UML Use Cases Diagram showing the UAV functionalities is presented in Figure 4.

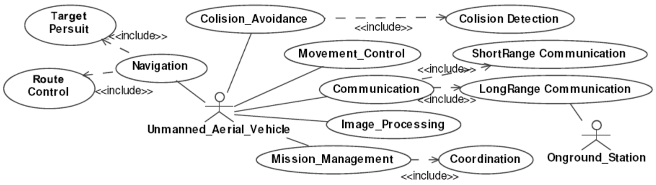


Fig. 4. UAV Use Cases Diagram

Based on the analyses of the UAV functionalities, a summary of the tasks that compose each subsystem is provided in the following paragraphs.

**Movement Control**: responsible to monitor and control the engines and direction mechanisms, such as flap actuators. It is composed by two tasks. The first is called `MovementController`, which performs the calculus that must be applied in the actuators and engines. The second task is called `MovementEncoder` and it is responsible for the sampling and encoding of the actual values in the engines and actuators, which will be used as feedback data for the `MovementController` task. These tasks are designed to have an implementation based on CPU, GPU, and PPU.

**Navigation**: control the directions of the UAV movements and sends control information to the Movement Control subsystem. It is composed by the `RouteControl` and `TargetPersuit` tasks. The first makes the calculation to guide the UAV through established waypoints, while the second performs the same, but for dynamic waypoints that varies accordingly to a moving object. These tasks have an implementation based on CPU, GPU, and PPU.

**Image Processing**: this subsystem gathers analog image information and performs its digitalization. It is composed of six tasks. The first is the `CameraController`, which is responsible by the movement of the camera, zoom and focus control of IRC and VLC, and antenna direction of the SAR. The second is the `Coder`, which codifies the analog input into digital data. The third is the `Compressor`, which compresses the digital images. The fourth is the `Reflectificator`, which is responsible for the reflection in the X and Y axis of radar image, as well as the rectification. These two processes are necessary to avoid distortions in the image. The fifth task is called `Filter`, which is responsible for filtering the radar images in order to eliminate the noise due to speckle effect [7]. The last task is called `PatternRecongnition` and is responsible to perform image segmentation and recognition of patterns from the previously processed data. These tasks are designed to have an implementation based on CPU, GPU, and PPU.

**Communication**: communication subsystem has two main tasks, the `LongRangeCom` and `ShortRangeCom`. The first provides connectivity with pair communication nodes in long distances, order of kilometers, while the second provides connectivity in short range, order of meters of distance. Both make use of a third, called `Codec`, which code and decode transmission data based on cryptographic techniques. These tasks have an implementation based on CPU, GPU, and PCICC.

**Mission Management**: this subsystem has two tasks, the `MissionManager` and the `Coordinator`. The first manages the information about the mission, like required data and mission policy, while the second drives the coordination with the other UAVs to avoid surveillance area overlap. Their implementation are based on CPU, and GPU.

**Collision Avoidance**: is composed by two tasks, `CollisionDetector` and `CollisionAvoider`. The first detects possible collisions with other UAVs of the fleet or non cooperative flying objects, and the second makes the calculus to avoid the collision and send them to the Movement Control subsystem. These tasks have an implementation based on CPU, GPU, and PPU.

### *F. Reconfiguration Approach*

In the presented experiment, the target architecture is composed of a four hybrid PUs platform: one CPU (Intel 2-core) and three types of co-processors, two GPUs (nVidia GeForce8800 GT – 512MB memory - using PCI Express x16 and CUDA-FFT/BLAS toolkit), a PPU (Ageia using PhysX SDK), and a PCICC (IBM using UDX toolkit). Figure 5 shows the desired execution platform, where the Profiling gathers information from the PUs and the Reconfiguration distributes the tasks along the PUs (intra allocation) and also consider sending data to be processed by other UAVs (inter allocation).

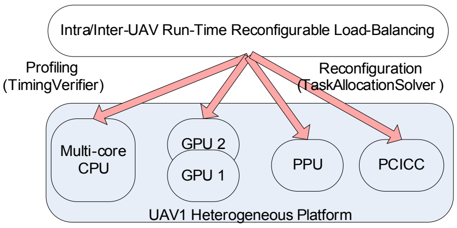


Fig. 5. UAVs Execution Platform with support for Load-balancing

Starting the mission, the UAVs have an initial task allocation throughout the CPU and co-processing units of the target platform according to one of the algorithms presented on sub-section III.A. In the current experiments, it was considered to use the ILP approach for the first distribution of the groups of tasks (and not the tasks individually). This simulation was done using the GLPK toolkit [8] and took in consideration the priorities of abstract tasks and its estimated execution costs in each PU, performing the following distribution: tasks from the Movement Control group (MovementController, MovementEncoder), and Navigation (RouteControl, TargetPersuit) are assigned to the PPU; the ones from Collision Avoidance (CollisionDetector, CollisionAvoider) are assigned to the GPU2; the Image Processing tasks (CameraController, Coder, Compressor, Reflectificator, PatternRecognition and Filter) are all assigned to the GPU1; and the sub-systems Communication (LongRangeCom, ShortRangeCom) and Mission Management (MissionManager, Coordinator) are firstly assigned to the CPU.

During execution, the mechanisms injected by the TimingVerifier and the aspects used by it - Jitter and ClockDrift - will start to generate values related to timing measurements. The Jitter and ClockDrift will take more time to generate more confident values, maybe after 100 executions in other to provide more meaningful measurements. But in an overall view, the TimingVerifier aspect will provide data to the TaskAllocationSolver, which will take data from the NodeStatusRetrieval and with the reasoning mechanisms that were inserted in the subsystems will analyze the data provided by these two aspects according to the algorithm introduced in III.B.

### *G. SAR Image Processing*

A special attention has to be given to the Image Processing sub-system. It is considered to be the group which requires more processing from the execution platform due to work with large data, being a key-factor to influence the dynamic reconfiguration of tasks. Figure 6 depicts the SAR Image Processing workflow. To summarize this figure, the captured data (brute scalar image) must be "adjusted" regarding the SAR position parameters (range and azimuth), followed by Fast Fourier Transforms (FFT) and image rotation and other corrections, to produce the final image. This process can be performed individually in the range and azimuth directions and it consists basically in a data compression on both directions using filters that maximize the relation between the signal and the noisy. Readers are addressed to [9] to get refined explanations about the workflow.

In terms of implementation, this sub-system can normally be developed based on CPU, but it fits better as a general processing on the GPU, since it involves mainly matrices multiplications applied to each scalar of the captured data. In addition, the data is represented using a complex number format (the real -32bits- and the imaginary -32bits- parts express the amplitude and phase of the scalar), generating large data ordering of gigabytes. Then, to optimize its execution time, a common approach is data partitioning. Individual regions can be processed in parallel by the available PUs (GPU, CPU, and PPU on this case) and, at the end, composed together to obtain the final SAR Image.

As a next step, the final image is submitted to a pos-processing phase, i.e., the PatternRecognition task aims to identify certain regions of interest that could contain objects specified in the mission directions as "pattern to be found". For that case, more resolution on those image parts will be needed and, consequently, new data will be generated, demanding more processing from the assigned PU(s) in order to produce the final images and extract new information (patterns). This scenario clearly influences the priority of tasks (old and new ones) since, at that moment, the new high-resolution images will have higher priorities comparing to others that became more "generic".

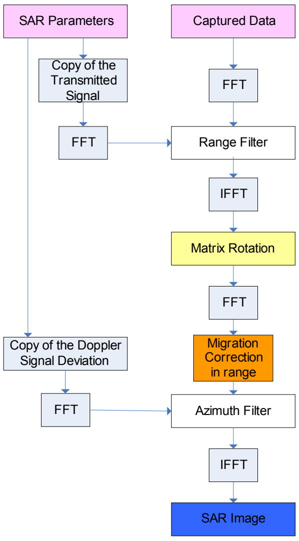


Fig. 6 – SAR Image Processing (based on the notes of [9] )

These events cannot be predicted a priory and the verification of such situation require, thus, a smart and dynamic reconfiguration support to reallocate the tasks, accomplishing the timing and requirements budget. On that case, the presented approach considers not just the balance of instructions inside an UAV execution platform, but also the data/image partitioning and interaction with other UAVs serving on the mission area. In this way, the reconfiguration support is applied to sending/receiving data to be processed by other idle UAVs and, at that point, encrypted communication must be applied in short and long range.

### *H. Previous results*

At this step of our research (reconfiguration and application architecture design), the UAV sub-systems algorithms, like Image Processing and Communication, were not implemented. The described scenario was simulated by means of creation of new tasks at run-time, where each task has an estimated cost to execute in each PU and a priority that change "on the fly". Table III exhibits the estimated costs and first priorities for the groups of tasks.

As the `TimingVerifier` aspect gathers online data about the execution of the tasks and the `NodeStatusRetrieval` gathers the PUs load parameters, the `TaskAllocationSolver` decides that the current allocation is possibly not the best configuration for the tasks that are waiting to be processed because it does not minimize the execution time. If confirmed, the reconfiguration takes place by using the `TaskMigration` aspect, which moves the tasks according to the decided new configuration presented by the `TaskAllocationSolver`. In the simplified simulation provided, evidences were gotten that when many new refined images are needed, the load-balancer tends to reallocate the Collision Avoidance tasks from the GPU2 to PPU (and then to CPU) and new instances of the Image Processing group (refined images) are assigned to be processed by GPU2 because of the Priority versus Estimated Cost compromise. In that situation, migration costs were estimated based on the throughput velocity of each PU bus (PCI, PCI-Express, etc.) and the other parameters considered on the equations (3) and (4).

Table III.
Estimation Costs for Tasks Groups

| Tasks Group | Estimated Costs (scale: 1 to 6) | | | | First Priority (scale: 1-6) |
|---|---|---|---|---|---|
| | GPU | PPU | CPU | PCICC | |
| **Image Processing** | 1 | 4 | 6 | - | 1 |
| **Collision Avoidance** | 3 | 2 | 5 | - | 2 |
| **Movement Control** | 2 | 2 | 3 | - | 1 |
| **Navigation** | 1 | 1 | 2 | - | 3 |
| **Communication Short/Long range** | 4/6 | - | 3/5 | 1/2 | 4 |
| **Mission Management** | 5 | - | 1 | - | 6 |

Moreover, as much as new refined images are required, it was verified assignment of tasks to other UAV that was idle, performing a drastic changing on the Communication task priority. The parameter ***L*** will, then, represent the communication latency between UAVs in short and long range. As it is reasonable to consider a grater ***L*** in long range than in short range, the simulation predicted that allocation in long range is not recommended and in most of the times the waiting time to access its PUs or other UAVs PUs in short range is worthwhile.

Based on that estimation and considering 2 UAVs, Table IV denotes the behavior of the dynamic reconfigurable load-balancer simulator.

The "first guess" represents one instantiation of each group of tasks that is assigned to a PU; and with the dynamic creation of new groups (4, 8, and 12 groups) of the Image Processing tasks, the assignment is changed and optimized, trying to minimize the total execution time. Note that these values cannot represent the best assignment since the simulator did not consider all parameters that influence the whole system. As it is an ongoing work, more accurate data about the reconfiguration will be provided along the refinement of the simulator in order to represent the scenario as reliable as possible.

Table IV.
Assignment of Tasks Groups for an UAV1

| Tasks Group | $1^{st}$ Guess | Dynamic Image Processing Created Tasks | | |
|---|---|---|---|---|
| | | 4 | 8 | 12 |
| **Image Processing** | GPU1 | GPU1<br>GPU2 | GPU1<br>GPU2<br>PPU | GPU1<br>GPU2<br>UVA2-<br>GPU1 |
| **Collision Avoidance** | GPU2 | PPU | CPU | CPU |
| **Movement Control** | PPU | PPU | CPU | CPU |
| **Navigation** | PPU | PPU | PPU | CPU |
| **Communication** | CPU | CPU | CPU | PCICC |
| **Mission Management** | CPU | CPU | CPU | CPU |

## V. Related Works

A set of tools named VEST (Virginia Embedded System Toolkit) [10] uses aspects to compose a distributed embedded system based on a component library. Those aspects check the possibility of composing components with the information taken from system models. This work also provides a library of aspects and has a type of model weaving, making different kinds of analysis to compose the system, such as schedule feasibility. However, it performs statically analysis at compiling time. In our presented proposal, aspects are used to change the system configuration at runtime, adapting its behavior to new conditions faced by the running applications.

Although there are some related works concerning dynamic reconfiguration in cluster computing, like [11] and [12], our approach concentrates on reconfiguration in off-the-shelf single PUs. This way, the work from [5] implements dynamic reconfiguration methods for Real-Time Operating System services running on a Reconfigurable System-on-Chip platform based on CPU and FPGA. The methods, based on heuristics and not on time measurements, take into account the idleness of the processing units and unused FPGA area to perform the load-balance.

In the field of programmability management, [13] gives an overview of the current programming models for multi-core processors, including the RapidMind API [14], which is a commercial tool that provides an interface to the programmer and abstracts specific co-processors development libraries, extracting parallelization automatically from its code. Besides, it supports load-balance in multi-core CPU, GPU, and the Cell, but, to our knowledge, not dynamically.

## VI. Conclusion and Future Works

Based on the need of non-functional parameters handling in modern applications and the advection of low-cost multi-core commodity hardware, this paper presents a new strategy of dynamic reconfiguration of tasks, involving aspect-oriented concepts to address the reconfiguration needs. It was presented real-time allocation and migration concepts applied to a modern heterogeneous execution platform.

An UAV surveillance system was used as case study and showed that modern application needs even more performance from "desktop" platforms, which are nowadays composed of several hybrid PUs. Real-time reconfiguration of groups of tasks was applied trough the UAV PUs and also considering the data sending to other UAVs.

Currently, we are defining even more suitable reconfiguration strategies and also working on finishing the implementation of the mentioned aspects that will effectively introduce the mechanisms to perform the reconfiguration in the real system. We plan to run and evaluate the tasks allocation in the platform of Figure 5 with algorithms that represents the real behavior of the tasks and according to the specific coprocessors. In the same way, we will work with the tasks individually and not just with its high-level groups and refine the strategy to allocate tasks among UAVs.

Finally, we intend to extract some knowledge about which task fits better on which platform according to different scenarios, i.e., distinct types of surveillance missions established to distinct UAVs platforms, improving the support on planning such missions.

## Acknowledgment

A. P. D. Binotto thanks the support given by DAAD fellowship and the Programme Alßan, the European Union Programme of High Level Scholarships for Latin America, scholarship no. E07D402961BR.

E. P. Freitas thanks the Brazilian Army for the grant given to him to follow the PhD program in Embedded Real-time Systems in Halmstad University in Sweden in cooperation with UFRGS in Brazil.

## References

[1] Kiczales G. et al. "Aspect-Oriented Programming", *Proceedigns of European Conference for Object-Oriented Programming*, Springer-Verlag, 1997, pp. 220-240.

[2] Freitas E. P., Wehrmeister M. A., Pereira C. E., Wagner F. R., Silva Jr. E. T., Carvalho F. C. DERAF: A High-Level Aspects Framework for Distributed Embedded Real-Time Systems Design. *In: Proc. 10th Int. Workshop on Early Aspects*, Springer, 2007, pp. 55-74.

[3] Wehrmeister M. A., Freitas E. P., Pereira C. E., Wagner F.R. Applying Aspect-Orientation Concepts in the Model-Driven Design of Distributed Embedded Real-Time Systems. *In: Proc. of 10th IEEE International Symposium on Object/component/serrvice-oriented Real-time Distributed Computing (ISORC'07)*, Springer, 2007, pp. 221-230.

[4] A. Burns et al. "The Meaning and Role of Value in Scheduling Flexible Real-Time Systems" *in Journal of Systems Architecture: the EUROMICRO Journal*. vol.46, n.4, 2000, pp.305-325.

[5] Götz Marcelo; Dittmann, Florian; Xie, Tao: Dynamic Relocation of Hybrid Tasks: A Complete Design Flow**.** *In: Proceedings of Reconfigurable Communication-centric SoCs (ReCoSoc'07)*, Montpellier, 2007, pp. 31-38.

[6] Stuart D. M. "Sensor Design for Unmanned Aerial Vehicles" *In Proc of IEEE Aerospace Conference*, 1997, pp. 285-295.

[7] Skolnik, Merrill I., Introduction to Radar Systems, McGraw-Hill, 3rd ed., 2001.

[8] The GNU Project, "GLPK – GNU Linear Programming Kit", http://www.gnu.org/software/glpk/, Jun. 2008.

[9] Cumming, Ian; Wong, Frank. Digital Processing of Synthetic Aperture Radar Data. Artech House-London, 2005.

[10] Stankovic, J.A. et al., "VEST: Aspect-Based Composition Tool for Real-Time System", in *Proc. of 9th IEEE RTAS*, 2003, pp. 58-59.

[11] Avresky, Dimiter; Natchev, Natcho; Shurbanov, Vladimir. Dynamic Reconfiguration in High-Speed Computer Clusters. *In Proceedings of the IEEE International Conference on Cluster Computing*, 2001, p. 380.

[12] Avresky, Dimiter; Natchev, Natcho. Dynamic Reconfiguration in Computer Clusters with Irregular Topologies in the Presence of Multiple Node and Link Failures. In *IEEE Transactions on Computers*, 2005, vol. 54, no. 5, pp. 603-615.

[13] McCool, Michael. Scalable Programming Models for Massively Multicore Processors. *Proceedings of the IEEE*, 2008, vol. 96, no. 5, pp. 816-831.

[14] McCool, Michael. Data-parallel Programming on the Cell BE and the GPU using the RapidMind Development Platform. *In Proceedings GSPx Milticore Application Conference*, 2006.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 589 – 592

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Student's Contest: Self-Driven Slot Car Racing

Milan Brejl
Freescale Semiconductor,
1. máje 1009,
756 61 Rožnov pod Radhoštěm,
Czech Republic
Email: milan.brejl@freescale.com

Jaroslav Nečesaný
Faculty of Electrical Engineering and Communication,
Brno University of Technology,
Purkyňova 118
612 00 Brno,
Czech Republic
Email: xneces01@stud.feec.vutbr.cz

***Abstract*—A recently announced university student's contest is based on a well-known entertainment – slot car racing. In contrary to the classical racing, here, the challenge is to build a car which can drive on an unknown track without any human interface and achieve the best possible time. This document describes the technical, algorithmic and educational aspects of a self-driven slot car development.**

## I. Introduction

THE proposal of the self-driven slot car contest was motivated by the idea to bring the university students into a development of a real-time software out of an isolated computer or laboratory environment. They would see the results of their work in real and could compare it with others on the contest. The contest subject is attractive. Probably every student knows and used to enjoy the slot car racing. Most of them think they know how to drive the car the best way.

It might look easy to drive a car which is guided and the only quantity to be controlled is the car speed. But it is not so. The development deals with a real-world compact standalone system, driven by a real-time software and using some kind of intelligence.

## II.Self-Driven Slot Car Principles

The development of the self-driven slot car requires a system engineering approach. The car mechanics, electrical hardware and the driving software need to fit together. The slot car mechanics is mostly fixed and done. For the electrical hardware a reference platform exists. The emphasis is taken to the software – the self-driving algorithms.

The key thing of a winning strategy is to learn the unknown track during the first lap and use the knowledge to achieve a maximum speed in the following laps. It might not be so easy to fully map the track during a single lap only. An adaptive process of getting better precision of the track parameters might be used during the whole run, enabling to get closer and closer to the optimum drive.

Before a self-driving algorithm can be implemented, the slot car electrical hardware needs to be built. The algorithm can run on an appropriate microcontroller, which has sensors connected to its inputs and a driver of the slot car DC motor connected to its outputs.

Regarding the sensors, the most suitable sensor for mapping the circuit is an accelerometer. There are easily accessible micro-electro-mechanical accelerometers in a small package, of a low weight and a low consumption. There is no rule which would restrict to mount for example a camera on the slot car, but one always needs to keep in mind its weight and processing demands. The slot car needs to race with all its equipment on.

Regarding the slot car DC motor drive, an integrated H-bridge or half-bridge power circuit is the easiest solution. These circuits usually have an integrated over-current protection and require just one pulse-width modulated signal to control the applied motor voltage.

In order to ease the student's development and let them focus on the slot car software, a reference hardware platform is available for reuse or as a starting point. This platform includes a choice of an 8-bit or a 32-bit microcontroller, a 3-axis accelerometer and a monolithic H-bridge.

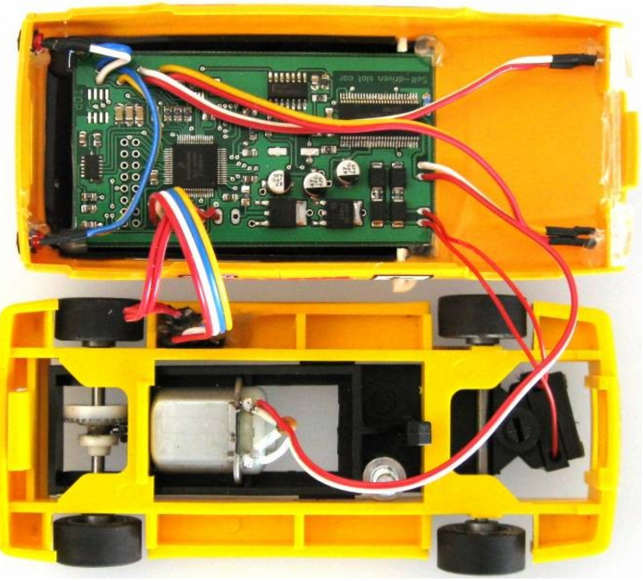


Fig. 1 The reference hardware platform built into a standard slot car

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## III. Contest Rules

There are three sets of rules. Rules of racing make the race clear in terms of the number of laps, measurement of time, start procedure, etc. These rules are strictly defined, so that everybody knows how his product will be measured. On the other hand, the rules for the track properties and car properties are let as free as possible. There is a set of rules unifying the slot car and track mechanical aspects, but the student's have their hands free to invent and implement various slot car improvements.

### *A. Rules of Racing*

Each contestant races separately against time. The slot car is placed on the track about 30cm prior to the lap counter. Then, the track is powered on. On the first pass through the lap counter the time measurement is started. The race is for a specified number of laps (e.g. 10) and the total time is measured. There are two race rounds and the sum of both race times determines the final results. The slot cars are placed to the right line of the track for the first round and to the left line for the second round.

The starting order is random for the first round. For the second round, the start order is based on the first round results. The contestant with the best time in the first round starts as the last one in the second round.

### *B. Track Properties*

The principal rule is that the race track is unknown to the contestants until just before the race, so that they can't adjust their slot cars for specific track parameters. Only the following track properties are specified:

- Track pieces producer
- Track length range (e.g. min. 10m, max 16m)
- Set of pieces the track can consist of
- Allowance of barriers
- Allowance of grade-separated junctions and altitude differences

The track properties are planned to progress year by year. For example the set of track pieces is limited to straights and curves at the beginning. Later, the set will be expanded by lane changes and crossovers. These pieces, as well as the barriers and altitude differences, may require special algorithms or even hardware improvements to detect them correctly, but also brings benefits if correctly handled.

### *C. Slot Car Properties*

The slot cars are powered from the track. The track voltage is fixed to 12V DC. No communication between the slot car and a remote controller is allowed. The only exception is a one-way car monitoring. There might be up to one switch on the car, allowing choosing between two modes of operation. The car weight is not limited. The car size is limited such a way that the car must pass through a tunnel of defined inner height and width. The slot car chassis and guide blade must be standard. Traction magnets are not allowed.

## IV. Discussion About The Slot Car Intelligence

The following discussion concentrates on several important aspects which need to be very well examined during the intelligent slot car development. The goal is not to get to a conclusion or solution. That is the contestant's job. The object it to show what kind of doubts and real-world issues the developers need to go deeply into.

### *A. Track Mapping Algorithm*

The track mapping algorithm is mainly based on the accelerometer measurement results. Theoretically, the integration of an instantaneous acceleration determines the actual speed, and integration of a speed results in a position. This way the track shape could be mapped. Practically, the car vibrations, sensor output noise, or a small DC error on the accelerometer output overpowers the useful signal after some time of the double integration. Hence, a reasonable approach is to remember the track as a time sequence of centrifugal accelerations, which is measured at a slow speed of the first lap, and which can be converted to a high speed driving scheme for the additional laps.

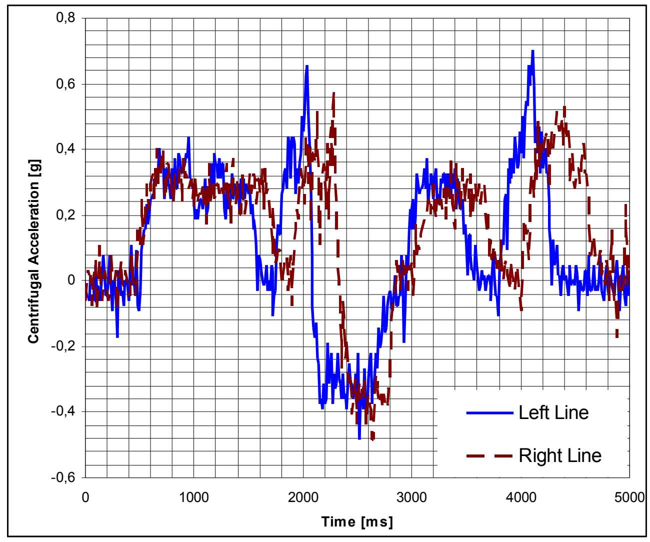


Fig. 2 A track and corresponding centrifugal accelerations measured by the slot car accelerometer

The most critical point is to recognize where the first lap ends and the second lap begins. After passing the minimal track length, the later measured part of the acceleration sequence can be compared (correlated) with the beginning part of the sequence. Once a match is found, the centrifugal acceleration sequence of the whole track is known, centrifugal forces can be predicted and the slot car can speed up.

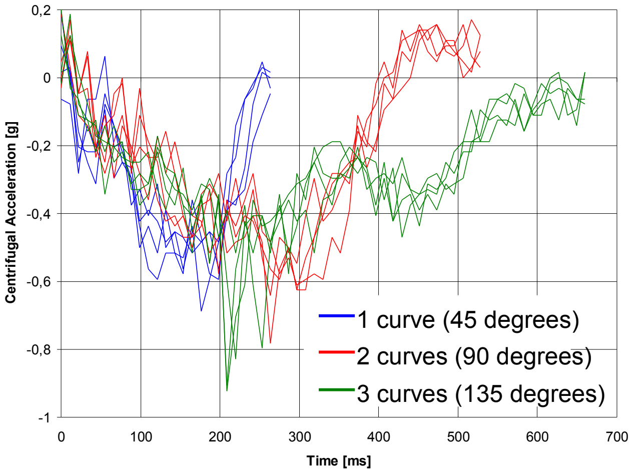


Fig. 3 Comparison of centrifugal acceleration courses measured in the smallest diameter curve of 3 different lengths (4 times each)

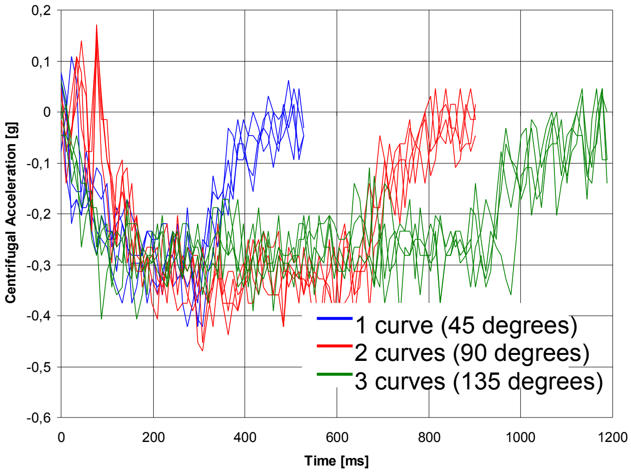


Fig. 4 Comparison of centrifugal acceleration courses measured in the biggest diameter curve of 3 different lengths (4 times each)

The described algorithm does not use any knowledgebase. With a knowledge base, the self-driving slot cars can work as a small expert system. The track property rules define a set of applicable track pieces. With the knowledge of each piece acceleration sequence, the measured sequence can be mapped to a sequence of the applicable pieces. This makes the measured acceleration sequence more reliable. Moreover, the track shape can be continuously compounded piece by piece into an XY plane. The point where the track closes corresponds to the moment when the car starts to drive the second lap. This way it can be found much sooner.

The knowledgebase may also include detailed drive schemes for various sequences of track pieces. For example, a left curve can be driven slightly faster if it is followed by a right curve, compared to if it is followed by a straight. Small differences like this one can make the big difference between the winner and the others.

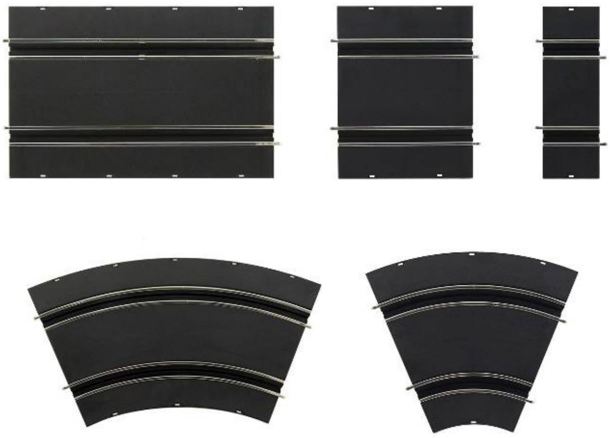


Fig. 5 Set of basic track pieces: straights and curves

An intelligent slot car might use track map information obtained during the first race round also for the second round, when the car goes in the other line. The lines are never similar, but a lot of important information of the right line can be transferred to the left line.

### B. Line Change and Crossover

The line change and crossover track pieces has interesting features. The most significant is the power supply gap. The slot car hardware needs to handle this in order to protect the microcontroller from a reset. On the other hand, the gap can be easily detected. It can serve as very reliable position information. There is always an even number of line changes within the track, usually two. It can be hardly confused with the crossover, because in case of the crossover there are two gaps shortly one after another. That is, again, a very reliable track mapping position feature.

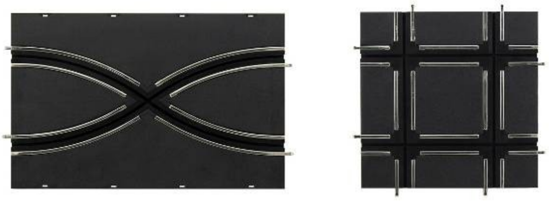


Fig. 6 Line change and crossover track pieces

### C. Barriers

The barriers might be not easy to detect with just the accelerometer. If reliably detected, the driving speed of the curve can be significantly faster.

### D. Altitude Differences and Grade-Separated Junctions

The track altitude differences, if allowed, bring the self-driving car several difficulties. The track mapping algorithm needs to be extended from mapping on an XY plane into mapping to an XYZ space. Also, on a flat track, the slot car speed is reasonably proportional to the applied motor voltage. This is not true on a track with altitude differences.

The car can map the altitude changes using the acceleration and tilt measurement using the Z-axis.

### *E. Slot Car Hardware Improvements*

The discussion above resulted into several hardware improvements required by some advanced algorithms.

The detector of a power supply gap can be a simple resistor divider and a low pass filter connected between the power input and a processor input pin. Note, that there are many short power supply losses and noise spikes caused by the movement of the power source braids and by their sparking, which should be filtered out preventing their detection.

A position sensor mounted on the DC motor shaft or an axle may bring significant advantages. The slot car speed can be measured and controller by a closed loop drive system. The information about the car position on the track, especially on a long straight track part, can be precisely calculated enabling to slow down as late as possible before the next curve. But even here are some limitations. Once the car goes into a skid, the tire speed is not equal to the car speed any more.

Many other hardware improvements could be identified. Addition of other sensors enables to process other quantities, but also requires handling the information fusion. The slot car weight and complexness might result in worse controllability and less robustness. It’s important to always question these aspects of hardware additions. On the other hand, the software changes are easily testable and there is a wide space open for the software improvements, for giving more and more intelligence to the self-driven slot car.

## V. Conclusion

The new contest should attract university student’s attention by the subject – slot car racing. This is a well-known and favorite entertainment all around the world. When the contestants start to work on the development of a self-driving slot car, with a motivation to win the contest, they discover wide possibilities how to improve the car intelligence and performance. On a deeper view, this brings them to handling real world issues in real-time software.

## References

[1] Competition declaration and rules, http://www.freescale.cz/.

[2] John Amos Comenius, *Schola ludus*, 1630, reprint *The School of Infancy*, University of North Carolina Press, Chapel Hill, 1956.

[3] Dave Chang, *Slot Car Handbook*, The Crowood Press, 2007, ISBN 978-1-861269-16-4.

[4] Robert S. Schleicher, *Slot Car Bible*, MBI, 2002, ISBN: 978-0-7603-1153-0.

[5] Robert S. Schleicher, *Slot Car Racing Tips & Tricks*, MBI, 2005, ISBN: 978-0-7603-2101-0.

[6] Czechia Racing Tack & SRC, http://www.autodraha-faro.cz.

[7] Scalextric, http://www.scalextric.com.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Real-time Support in Adaptable Middleware for Heterogeneous Sensor Networks

Edison Pignaton de Freitas
IDE – Halmstad University – Sweden / PPGC UFRGS - Brazil
Email: edison.pignaton@hh.se

Marco Aurélio Wehrmeister
PPGC UFRGS – Brazil
Email: mawehrmeister@inf.ufrgs.br

Carlos Eduardo Pereira
PPGC UFRGS - Brazil
Email: cpereira@ece.ufrgs.br

Tony Larsson
IDE – Halmstad University – Sweden
Email: tony.larsson@hh.se

*Abstract* — **The use of sensor networks in different kinds of sophisticated applications is emerging due to several advances in sensor/embedded system technologies. However, the integration and coordination of heterogeneous sensors is still a challenge, especially when the target application environment is susceptible to changes. Such systems must adapt themselves in order to fulfil increasing requirements. Especially the handling of real-time requirements is a challenge in this context in which different technologies are applied to build the overall system. Moreover, these changing scenarios require services located at different places during the system runtime. Thus a support for adaptability is needed. Timing and precision requirements play an important role in such scenarios. Besides, QoS management must provide the necessary support to offer the flexibility demanded in such scenarios. In this paper we present the real-time perspective of a middleware that aims at providing the support required by sophisticated heterogeneous sensor network applications. We propose to address the real-time concerns by using the OMG Data Distribution Service for Real-time Systems approach, but with a more flexible approach that fits in the heterogeneous environment in which the proposed middleware is intended to be used. We also present a coordination protocol to support the proposed approach.**

## I. Introduction

SENSOR network applications are becoming more complex due to the use different kinds of mobile and sophisticated sensors, which provide more advanced functionalities [1] and are deployed in scenarios where context-awareness is needed [2]. To support those emerging applications, an underlying infrastructure is necessary. The current proposals suggest the use of a middleware, such as TinyDB [3] and COUGAR [4]. The main drawbacks of these state-of-the-art middleware are the following assumptions: (i) the network is composed only by a homogeneous set of basic or very constrained low-end sensors; (ii) the lack of intelligence of such network that compromises the adaptability required facing changing operation conditions, e.g. lack of QoS management and control. Adaptability is a major concern that must be addressed due to: (a) long deployment time of wireless sensor networks may require flexibility in order to accomplish changes in the requirements during usage life time of the network; (b) wireless sensor networks are deployed in highly dynamic environments, implying that applications have to be flexible enough in order to continue being used in these scenarios. In such environments, real-time requirements are especially hard to be met, because of variable operational conditions, and thus there is a need of adaptation of real-time parameters to operational conditions. QoS management must therefore be flexible, allowing renegotiation among nodes during the system runtime [5].

This paper reports a work in progress related to the development of an adaptive middleware to support sophisticated sensor network applications that must adapt its behavior according to changes in the environment and the application demands. We use the concept of multi-agents to provide the reasoning about the network and, besides other things, to decide about time-related requirements and QoS control. This paper focus in the real-time features itself without considering how the multi-agents perform the reasoning. Based on the main real-time concerns that affect heterogeneous sensor networks, we propose the use of mechanisms to address them supported by a coordination protocol. The main contributions provided by this paper are the description of the proposed handling of the outlined real-time concerns by means of the proposed mechanisms and coordination protocol. Besides, the overall middleware proposal consists in a contribution by providing a flexible variant of a middleware for heterogeneous sensor networks based on an OMG standard.

The remaining of the text is organized as follows: Section 2 presents an overview of the proposed middleware. Section 3 discusses some main related real-time issues in middleware for wireless sensor networks. Section 4 presents the proposed approach to support real-time requirements in the proposed middleware. Section 5 presents the coordination protocol that partially supports the proposed approach. In Section 6 some related works are outlined, while Section 7 gives some concluding remarks and directions of the future work.

## II. Overview of the Proposed Middleware

The general idea is to develop a flexible middleware that can be used to support applications in heterogeneous sensor networks. In the context of this paper, heterogeneity means that nodes in the network may have different sensing capabilities, computation power, and communication abili-

This work is partially supported by KK Foundation.

ties, running on different hardware and operating system platforms. The main goal is that the proposed middleware fits both low-end and rich sensors. In order to achieve this goal, aspect and component oriented techniques will be used in a way similar to the approach presented in [6][8] and the mobile multi-agents approach [9].

Low-end sensors are those with simple capabilities, such as piezoelectric resistive tilt sensors, needing limited processing support and communication resource capabilities. Rich sensors comprehend powerful devices like radar, visible light cameras or infrared sensors that are supported by moderate to high computing and communication resources. Thus, in order to deal with these very distinct capabilities, the proposed middleware must be lightweight, while being scalable and customizable. For instance, it might handle the node's resource usage in order to assist tasks distribution among different nodes that are capable to accomplish them. Another feature of the middleware is to help provide the quality of the data required by a certain user's demand, such as accuracy and precision. Better results can be achieved by choosing the correct set of sensors to perform measurements and data collection. The mobility characteristic is also related to the heterogeneity addressed by the middleware. Sensor nodes can be static on the ground or can move themselves on the ground or fly over the target area in which the observed phenomenon is occurring. The Fig. 1 graphically represents the idea of the heterogeneity dimensions considered in this work.

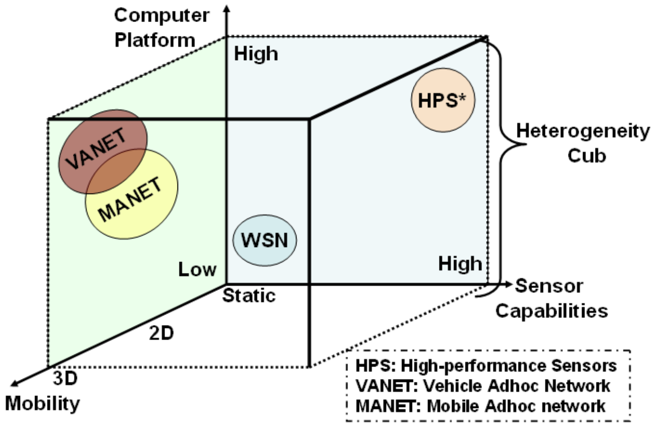


Fig. 1 Heterogeneity Dimensions

The input to the sensor network system, coordinated by the proposed middleware, is seen as a "mission" that the whole network has to accomplish. In order to allow that, a high-level Mission Description Language (MDL) is being formulated based on the C/ATLAS test language [10]. This language will allow the specification of data (at a high level of abstraction) in which the user is interested, including constraints regarding timing and location limits as well as the measurement rate or accuracy desired. This high-level user information will be translated to system parameters, such as QoS related parameters. The proposed language will also allow hierarchical description of the mission goals, with establishment of priorities and other details, for instance, the application of comparison metrics to evaluate how well the mission is being accomplished. The priorities given to missions will also affect QoS parameters. Consequently, the middleware must handle them in order to fit QoS parameters according the established priorities for each task. As an example of usage of the MDL, the user may want to gather data about a certain kind of vehicle that passes in a specific area during a given period, with a high accuracy level. In order to do this, it will specify the target area, a distinguish characteristic of the vehicle (e.g. over a certain weight or with a certain shape), provide the period during which the mission will be performed and how accurate results he or she desire from the system. This high-level information is though translated in internal parameters that will drive the network to perform the mission.

The middleware must perform its actions also in very dynamic and changing scenarios. Thus the set of sensors chosen in the beginning of a mission may not be the most adequate one during the whole mission. For example, an area surveillance system receives the mission to survey an area that may not allow traffic of certain kinds of vehicles. Ground sensors are set to alarm in the presence of unauthorized vehicles. Additionally, Unmanned Aerial Vehicles (UAV) equipped with visible-light cameras is set to fly to the area where a ground sensor has issued an alarm to verify the occurrence. However, a sudden change in the weather, e.g. the area becomes foggy or cloudy, turns the use of a visible-light camera useless. This type of change in the operational conditions must be supported by the middleware, which must be able to choose a better alternative, among the set of available options, for instance by choosing an UAV equipped with an infrared camera instead.

The dynamicity of the operation scenarios may force the adaptation of system's real-time parameters in order to accomplish a certain mission. As an example, certain data forwarding traffic may overload a node in the path from the data gathering points to the final user. Incoming data can experience problems like undesired delay or unpredictable jitter, so solutions as data flow priority assignment and/or use of another node as forwarder may take place. Priorities and choice of alternative paths may further not be static from the start of the system runtime, but can dynamically be changed according the user requirements and changes in the network such as node failures.

The middleware is divided in three parts or layers indicating that they are partly using each other in a specific order. Fig. 2 presents the overview of the layers of the proposed middleware, and a description of each layer is provided as follows.

The bottom layer is called *Infrastructure Layer*, which is responsible for the interaction with the underlying operating system and for the management of the sensor node resources, such as available communication capacities, remaining energy, and sensing capabilities. This layer also coordinates the resource sharing based on application needs passed through the upper layers. Services provided by upper layers may need some resource sharing support, which is encapsulated in the infrastructure layer. As an application uses such a service, the corresponding layer asks for the infrastructure layer to manage the access control to the required resources.

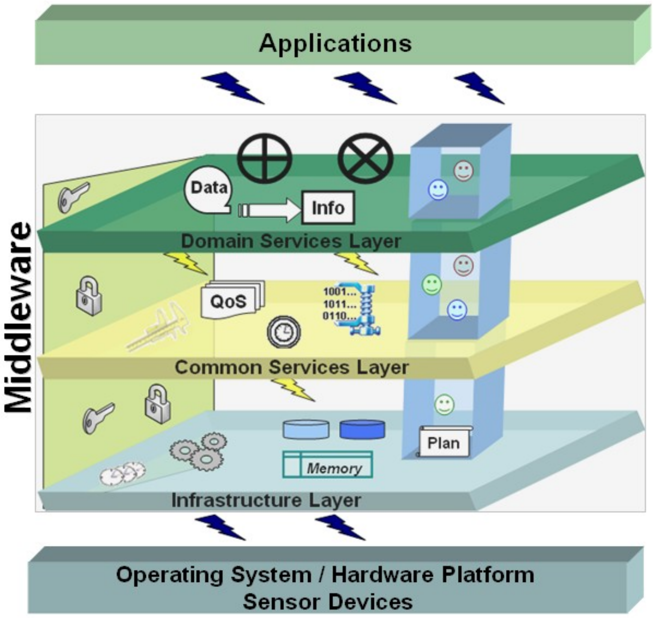


Fig. 2 Overview of the Middleware Layers

The intermediate layer is called *Common Services Layer*, which provides services that are common to different kinds of applications, such as QoS negotiation and control, quality of data assurance, data compression and the handling of real-time requirements, including storing of parameters. Other concerns such as deadline expiration alarms, timeouts for data transmissions, number of retries and delivery failure announcements, resource reservation negotiation among applications (based on priorities and operation conditions), dynamic bindings, synchronous and asynchronous concurrent requests are also handled by this layer.

The top layer is called *Domain-Services Layer* and has the goal to support domain specific needs, such as data fusion support and specific data semantic support to allow the production of application-related information from raw data processing. Fuzzy classifiers, special kinds of mathematical filters (e.g. like Kalman filter) and functions that can be reused by applications of the same domain will be found in this layer.

Multiple applications can run concurrently in the network. The middleware handles resource sharing and provides data sharing among applications that need the same type of data, allowing a better energy use in resource constrained nodes. In powerful nodes, with more energy available, the middleware can provide more complex services aiming at the handling of rich data, such as those related to image processing, and pattern matching. This also means that such nodes can take some of the burden from meager nodes.

"Smile faces" in the Fig. 2 represent agents that can provide specific services in a certain node at a certain moment of the system runtime. A special region (called *agents-space*) links them throughout the layers, allowing the exchange of information. The *Infrastructure Layer*, in the bottom, has just one agent, which is responsible for planning and reasoning activities. Interested readers can find more information about the use of agents in our middleware in [9].

Concerns that affect elements in more than one layer of the middleware, such as security, are represented as cross-layer features. In Fig. 2, the "locks" and "keys" in the left-side plan represent this idea for the security concern example. These crosscutting concerns will be addressed in our middleware with the aspect-oriented approach presented in [6] and [7]. Dark lightning bolts represent the communication activities between applications and the middleware in the upper part of the Fig. 2 and between the middleware and underlying operating system and hardware (including sensor devices) in the lower part of the Fig. 2. Light lightning bolts in the middle of Fig. 2 represent communication among internal elements of the middleware.

## III. Key Features for the Wireless Sensor Network Middleware

The proposed middleware is intended to be used in dynamic harsh environments. Changes in this kind of scenario occur frequently, requiring system adaptability. However, it is not useful to provide adaptability without guaranties that the desired data will be delivered in time. So, the timing related concerns about the network consist also in an adaptable dimension that the middleware must take into account in order to successfully provide the necessary support to applications running within those scenarios. With this thought in mind, some key features that the middleware for wireless sensor networks must take into account are presented [11] [12]:

- **Avoid Single Point of Failure:** no single node must centralize any type of registration or submission service for the entire network, as in broker-based architectures. This concern is related not only with the possible node fail, but also with node overload and incapacity to respond the demands of all clients;
- **Dynamicity in System Structure and Network Topology:** as operation conditions change, new nodes can come in and out. Moreover, different services can be required in different places at different times;
- **Reduce Data Communication:** As large amounts of data may need to be exchanged among nodes, the bandwidth must be carefully used to avoid congestions and flooding;
- **Meet Applications Requirements:** Applications must meet timing requirements, based on their users' needs for data, and on the conditions under which they want the data. These requirements may change during runtime, what requires a fine tuned QoS management and control;
- **Real-time Requirements:** The middleware must provide the specification of real-time requirements. At the same time, it must handle the translation of these requirements into specific QoS parameters, in order to accomplish the modification of this high-level requirements;

- **Fault-Tolerance:** Nodes may fail, links can go down and network errors occur, but the application that requests data cannot wait indefinitely. Mechanisms of errors reporting must be provided, as well as alternative resources may be offered. Activities such as time-bounded activities monitoring, delays and deadlines management must be handled in order to provide dependable support.

These above highlighted issues crosscut several concerns, i.e. they are intertwined with several characteristics. For example, QoS values, such as deadlines and bounded time actions/delays, impact in the management of communication and fault-tolerance. The resulting complexity indicates a need to use proper abstractions in order to deal successfully with these concerns.

## IV. Addressing Key Issues of Sensor Networks with the Proposed Middleware

The proposed middleware is inspired in the Data Distribution Service for Real-time Systems (DSS) specification, standardized by OMG [13]. The proposed approach is based on the publish-subscribe paradigm. Some nodes publish their capabilities and the offered data, while others subscribe to data, in which they are interested.

Although being inspired on the OMG DSS standard, the middleware does not follow the whole specification. As it is intended to fit both low-end nodes (based on simple and constrained platforms) and sophisticated ones, it must not only be lightweight but also provide capabilities for customizations in order to deal with the needs of the sophisticated sensors. Consequently, the middleware uses a minimalist approach, keeping it as simple as possible in each node. It may be considered as a "Lightweight Flexible DSS-based Middleware", in which the handling of some features will be presented in a soft version, such as the handling related to topics described in the ***Domain Module*** section of the DSS specification [13]. This soft version will simply data structure and check mechanisms that will be deployed middleware instance for low-end nodes, for instance.

Additionally, features can be included or modified in the middleware by using adaptation and extension mechanisms, such as the inclusion of additional components, as well as the weaving of specific behaviors by aspects. As an example of more explicitly of how the aspects will be used, the handling of QoS policies, which affect different elements in the system, can be concentrated in a aspect avoiding the spread of this concern over different classes, and like this, making it easier to promote maintainability and evolution of those policies. In the DSS specification, several classes (e.g. , `DomainParticipantFactory`, `Topic`, and `DomainParticipant`) have parameters and methods to handle the QoS policy concern. Using aspects, the behaviors encapsulated in those methods spread over several classes can be concentrated in one aspect that weaves classes in the system accordingly the policy needs. Other proposed adaptations in the DSS original specification are: (1) the simplification of the status model, which will consider a subset of the original proposed object inheritance tree presented in section dedicated to the details about communication status in [13]; (2) adaptation of the cache model in order to add the cache usage proposal that will presented further in this section.

The following subsections present how the proposed middleware will address the needs presented in Section 3.

### *A. Flexibility*

The middleware provide full control of the communication, it does not use underlying control mechanisms available in the nodes' network layer. Instead, it provides its own communication control. It means that all parameters related to communication are controlled by the middleware, using only basic connectionless communication services offered by the nodes' network layer. The middleware handles parameters like number of retries, message priority, memory usage for buffering and timing. This control over communication provides more flexibility to manage the messages exchanged by each node, with direct impact in the reduction of latency. Moreover, it gives a finer grained control if compared with the simple use of the communication primitives offered by the native nodes' operating system network services.

### *B. Dynamicity*

Using the publish-subscribe paradigm, when a node gets into the network, its services are announced and the interested nodes subscribe for them. This eliminates the need for a dedicated server node that centralizes the available services in the network. Additionally, it reduces latency in acquiring data because there is no intermediary node between the data producer and the consumer.

### *C. Minimum Message Exchange*

Using the publish-subscribe paradigm by itself already reduces the number of exchanged messages, due to the elimination of intermediate nodes, such as request-brokers. However, the use of bandwidth for control messages still exists. It can be reduced with the use of a smart protocol to avoid unnecessary messages exchange, as explained with details in the following section. The protocol is called CUME (Cut Unnecessary Messages Exchange).

### *D. Multicast Communication*

The middleware use a multicast communication to reach selected destination nodes. For instance, the publisher sends its data only to the nodes that subscribed to it. This type of communication affect positively the latency and throughput, as data is sent at the same time to several nodes without unnecessary broadcast and without delays that would occur if a unicast communication was used; since then the publisher node would have to resend the same data several times, one for each subscriber. A negative-acknowledgement (NACK) strategy is adopted in order to reduce acknowledgement messages in the network. However, very sensible data may require a positive acknowledgement in order to assure its delivery. To address this need, positive acknowledgement is also available and can be used when required. This acknowledgment strategy is also part of CUME.

### *E. Network Resources Usage Control*

The control of the use of the communication media and transmission buffers are crucial in order to improve the overall system performance. The middleware perform this task by taking into account two factors: (i) the priority associated to each application; and (ii) the resource sharing policy adopted in the system. There are three available resource sharing policies:

- **Fair Sharing:** the priorities are not considered and thus all applications have the same right to use the resources in a round-robin scheme, which is organized in a incoming FIFO queue;
- **Soft Priority Sorted** : the priorities are taken in account. However, if a higher priority application needs to use a resource already used by a lower priority one, it must wait until the later release the resource. Due to its higher priority, it will get access to the resource before other applications, which may be waiting for the resource;
- **Mandatory Priority** : higher priority applications can preempt lower priority applications in order to access the desired resources.

### *F. QoS Control*

The QoS control is done through a contract between the data provider and the data requester. When a node publishes a data service, it informs also the QoS level that it is capable to offer. Nodes interested in the published data service those accept the offered QoS level, subscribe to the service. However, if a node is interested in the data but does not agree with the offered QoS, it has two alternatives:

- If the application that is requiring the data has a priority lower than the others using the same service, it looks for another data provider;
- If its priority is higher than other applications, it negotiates with the data provider node, in order to obtain the desired QoS in spite of the bad consequences that it may imply to other lower priority applications.

As an example, a node may provide a certain type of data (D-1) at each 10 milliseconds and another type of data (D-2) at each 25 milliseconds. The first, D-1, has two subscribers and the second one, D-2, just one. A forth node wants to receive D-2, but at each 10 milliseconds. The data provider node does not have the available resources in terms of processing and communication power to deliver both data at each 10 milliseconds. It can deliver just one type at the desired rate. If the forth node has an application with a priority higher than the applications running in the other nodes, it will negotiate with the provider and, if it is feasible, the provider will change its behavior to accomplish the need of the requesting node with the highest priority. If it is not feasible, the requesting node will look for another provider.

### *G. Use of Cached Values*

Some measurements aim to gather information about changes in values of certain observed phenomena. The use of cache in both data providers and requesters may avoid unnecessary data communication. When the measurement device gathers a new value, the data provider updates its own cache and publishes the new value updating its subscribers. If the size of the data is large and requires several packets to be transmitted, a differential value can be send instead of the whole data value, for instance, using just one package. This differential value will be used to update the current value stored in cache. The use of this option is arranged in advance at the time when the nodes are negotiating the QoS contract.

### *H. Fault Tolerance*

In order to support the use of strategies like cached values and to detect node failures, fault tolerance mechanisms must be provided. A heartbeat mechanism is used when a node does not have any data to send. Thus it broadcasts a message in order to inform other nodes that it is still alive. It works well for nodes in the range, but for those not in the range and that are interested in the provider, directed heartbeats must be sent. The periodicity of the heartbeat sending is a configurable parameter. Another fault-tolerance support is a list of possible backup service providers. When a node that provides a certain kind of data service stops working, nodes interested in that data look for another node at the backup list that provides the missing data.

### *I. Network Partitioning*

Network partitioning can occur intentionally or by uncontrolled conditions. The first one occurs when groups of nodes have much communication among each other and little outside the group. Thus, they form a cluster in order to minimize the communication with outside nodes. A cluster-head, which is responsible for communications with clusters outside nodes, is elected. This election is based in the actual status of the involved nodes, concerning remaining resources and quality of the communication link. The use of a cluster-head does not characterize a single point of failure because any node in the cluster can be elected. If the current cluster-head fails, a new one is elected. The cluster-head formation is used only if the QoS requirements are met. For instance, if the presence of the intermediary communication with the cluster-head impact latency, a communicating node can leave the cluster. If it has a specific need for a data that is not interesting to the other nodes (or have different QoS requirements) it can make a contract directly with the provider of the required data, without passing the cluster-head. This way the cluster of nodes is not rigid, but flexible.

Uncontrolled conditions, such as node failures, communication obstacles and interference can also promote network partitioning. This kind of partitioning is not desirable and planned such as mission related clustering, and to handle this issue, isolated nodes store as much data as possible to send when the link be reestablished, what can be done by the deployment of new nodes, by the mobile nodes that come into the area or by the disappearance of the obstacles or interferences that caused the partitioning.

### *J. Data Segregation*

There are two kinds of data exchanged among nodes in the network: control data and application data. Control data is small and may not experience latency or unexpected delays to achieve its destination. So, control data is segregated from application data by receiving higher priority to be forwarded. On the other hand, there are several kinds of application data, e.g. simple values (integers and floats), video stream and character string. In spite of this sort of data have a priority lower than control data; they must fulfill the QoS requirements of the application. Moreover, jitter is also reduced by the segregation, because segregated data follows different buffers.

### *K. Synchronous and Asynchronous Calls*

The middleware is intended to support both synchronous and asynchronous calls. Synchronous calls are bounded in time in order to avoid unpredictable waiting periods by the caller applications. The waiting time and number of retries are configurable. In case of the expiration of the waiting time (timeout), or if the number of retries is reached, specific handling mechanisms can be triggered. Asynchronous calls are also provided and they are used, among for other proposals, to support the handling of asynchronous events.

## V. CUME Protocol

CUME (Cut Unnecessary Message Exchange) is a protocol that aims at minimizing the number of control messages exchanged among nodes, and additionally to optimize the use of data messages.

The publisher-subscriber approach proposed is slightly different from the state-of-the-art middleware available. Instead of announcing its service when coming into the network, a node does that in different manners depending on the situation, on the type of the node and data provided.

For instance, let's consider moving sensors embedded in UAVs, which fly in a cluster formation to accomplish a mission in a certain place (passing through several way-points). The group of UAVs can find another UAV flying in the same direction, which we call an alone-UAV. In this case, the protocol will perform the following: (i) exchange nodes' destination information; (ii) exchange capabilities and mission details information; and (iii) negotiate the use of the nodes' resources if it is the case. In the following each of these phases are explained with more details.

The alone-UAV just tells next way-point to the cluster-head of the UAV-cluster. Then the later sends a message with the next way-point destination to the cluster. If the destinations are the same, the alone-UAV joins the group without extra confirmation messages. There is no acknowledgement message, if one of these sent messages was lost, the other node that was expecting to receive the message send a NACK requiring retransmission. The waiting time for sending the NACK is estimated by the distance between the nodes, and then, a timeout is set in accordance.

Being in the cluster, the alone-UAV can use short range communication to announce its services and the mission that is supposed to accomplish. If there are common interests among the alone-UAV and the UAV-cluster, they cooperate. If there are not, they analyze the capabilities of each other and recognize if any of them is needed for its own mission and if it lacks this capability. Based on the established priorities and the mission conditions, they will decide which mission is more important and the resources will be allocated to that mission according to the priorities. This is an example in which unnecessary long-range communications were avoided.

In the case of different destinations, they will perform the exchange of the capabilities information anyway using long-range communication to make the same reasoning described above.

The CUME protocol in addition uses caches in the nodes to avoid unnecessary messages exchange. If a publisher has two subscribers of a certain data; one that requires the data at each 10 milliseconds (called node A) and another at each 20 milliseconds (called node B). After the data provider sends the first value for each, it stores the value in its local cache. Nodes A and B receive the data, giving it to the requesting applications and storing it in their caches. After 10 milliseconds, the provider compares the measured value with the one stored in its cache and, if it is different, sends it to node A, updating its cache. However, if the value is the same, it just sends a heartbeat to node A, which understands that the value is the same and gives the value in its cache to the requesting application. In the next 10 milliseconds, the same occurs, but the update or the heartbeat will be sent to both requesting nodes A and B. In this situation there is no acknowledgement for each message. If an expected message is lost, a NACK is sent from the consumer nodes to the data provider. In order to avoid a NACK storm in the case of several requesting nodes miss the sent data, a random timer is set in each node that loosed the message to send the NACK. The value that is set in the random timer is chosen in a range according to the established QoS requirements. When a value must be updated, it is possible that just the difference between the new and the old value is sent. This is also a parameter exchanged during the negotiation.

Caching negotiation is also part of the CUME protocol. It is done by an exchange of control information that will represent the agreement about the freshness time of the cached values, and the accepted delay to receive the refresh of the cached data or a new value. Besides, the cache space is not a fixed amount of memory but it is negotiated. These parameters can be renegotiated according to the needs and/or changes in the environment and operation conditions. Threshold values for renegotiations are established in the first negotiation round, in order to avoid unnecessary exchange of control messages due to small changes, which do not represent a real need for renegotiation. The parameters will depend on the application and the type of mission, thus a flexible set of parameters can be used according to the individual needs presented in different situations.

The CUME protocol is also responsible for monitoring the rate of messages forwarded by the nodes. If the node's throughput is arriving to its limit, the node informs the sending nodes to use different nodes to forward their messages, according to the priorities of the applications that require the communication. The limit of bandwidth usage is also a configurable parameter that can be adapted according to the operational conditions, such as interferences or amount of collisions.

## VI. Related Works

MiLAN [14] (Middleware Linking Applications and Network) is an adaptable middleware that explores the concept of proactively adaptation, in order to respond the needs in terms of QoS imposed by changes in the operational environment. The authors claim that a major drawback in the most of existing middleware for sensor networks is the fact that they just provide reactive adaptation. In dynamic environments as those in which wireless sensor networks are deployed, this behaviour does not cover the real needs for adaptation required by running applications. MiLAN allows the specification of the required QoS for data, adjusting the network to increase its lifetime, by efficiently using energy, in the same time that the quality needs are still meet. The major difference in relation to our approach is that MiLAN does not provide a customization mechanism to enrich the middleware features and support more sophisticated sensor nodes like those carried by UAVs. This difference points to a drawback indicating that MiLAN is not prepared to manage high intense data flow generated by traffic of images or video streams.

Atlas [15] is an architecture for sensor network based on intelligent environments. The main goal of this architecture is to provide services and support to applications like health care assistance in intelligent houses equipped for assist the elderly. The Atlas middleware is based on the OSGi Service Platform Specification [16], which is a framework that provides a runtime environment for dynamic and transient service modules called Bundles. This middleware offers some support to real-time, but as a secondary feature derived mainly from the adaptable service discovery feature and Bundles specification. However, this support is too simple and do not promote the desired control of real-time parameters as we proposed in our approach. Furthermore, the Atlas middleware does not fit into low-end nodes with constrained resources.

Real-time CORBA (RT-CORBA) [17] is a successful middleware that provides real-time support for distributed applications. It is based on the CORBA standard [18], which introduces a priority mechanism to map native operating system priorities into remote nodes priorities. It also provides predictability by managing resource allocations according to the established priorities. The use of thread pools prevent the unbounded blocking periods based on the resource usage by the lower priority applications, and priority inversion. RT-CORBA fits well in sophisticated sensors nodes with a rich computing platform, but it is too heavy to run in low-end nodes. In our approach instead, the proposal is to address the real-time needs in both low-end and rich sensor nodes with different computing platforms and available resources.

Quality Objects (QuO) [19] proposes the addition of a QoS adaptive layer on existing middleware, such as RT-CORBA. It provides means of specify QoS requirements, monitor and control the provided QoS, and also adapt the middleware behavior according to the QoS variations that may occur during runtime. This proposal presents an interesting approach to support those operations using: (i) *contracts*, which encloses the QoS requirements; (ii) *delegates*, which are proxies that can be inserted into the path of object interactions transparently to weave the QoS awareness and adaptive code; and (iii) *system condition objects*, which provide consistent interfaces to infrastructure mechanisms. However, as this framework relies on an existing middleware such as RT-CORBA, it has the same drawback indicated above, i.e. it cannot be used in low-end nodes.

## VII. Conclusion and Future Works

This paper presents the real-time support offered by an adaptable middleware for heterogeneous wireless sensor networks. Real-time concerns, which affect a network composed by heterogeneous sensor nodes and impose difficulties on the handling of them, are handle through the proposed middleware that fits both rich sophisticated nodes and low-end constrained nodes. Our proposal presents a protocol to cope with the unnecessary control message exchange, and use different mechanisms to address real-time issues. Besides bandwidth savings, it has the side effect to increase the security against eavesdroppers, by diminishing the number of exchanged messages. The use of mechanisms like caching and differential updating is also provided in order to diminish the bandwidth usage.

Related works in the area do not address both types of sensor nodes. The majority of middleware for sensor networks consider a homogeneous network composed by low-end nodes, producing very simple data, like the approach presented by MiLAN. In the other extreme there are middleware that consider a network composed by powerful sensor nodes that, in some cases, do not even have any concerns about energy consumption, as the assumption presented by Atlas. Conversely, the proposed middleware addresses both low-end and rich sensor nodes, using mechanisms that provide support to both simple and sophisticate data.

As future works we are refining the mechanisms of the CUME protocol in order to incorporate different strategies to provide a better response to problems like jitter. The coordination mechanisms are also under development. In this paper we gave an overview of the type of coordination that it will be provided, as presented in example of the UAV-cluster meeting the alone UAV. However, we are working to provide similar kinds of coordination in other situations and with different kinds of sensor nodes. Moreover, we still have to simulate a complete case study scenario and assess the measurements in order to validate our assumptions before the final implementation; and also concerning the implementation, proposals of use composed link metrics in the message forwarding decisions (routing) [20] and probabilistic tendencies for the cluster-head election [21] are being considered.

## Acknowledgment

E. P. Freitas thanks the Brazilian Army for the grant given to him to follow the PhD program in Embedded Real-time Systems in Halmstad University in Sweden in cooperation with UFRGS in Brazil.

## References

[1] D. Culler, D. Estrin, and M. Srivastava, “Overview of sensor networks” , *IEEE Computer* , vol. 37, no. 8, pp. 41–49, 2004.

[2] K. Henricksen and J. Indulska. “A software engineering framework for context-aware pervasive computing”, *In 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom),* pages 77–86. IEEE Computer Society, March 2004.

[3] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. “TinyDB: An acquisitional query processing system for sensor networks”, *ACM Transactions on Database Systems* , 30(1):122–173, 2005.

[4] P. Bonnet, J. E. Gehrke, and P. Seshadri. “Towards sensor database systems”, *In 2nd International Conference on Mobile Data Management (MDM)* , volume 1987 of Lecture Notes, 2001.

[5] V. Liberatore. “Implementation challenges in real-time middleware for distributed autonomous systems”, In *Prof of Second IEEE International Conference on Space Mission Challenges for Information Technology, 2006. (SMC-IT 2006)*.

[6] E. P. Freitas , M. A. Wehrmeister, C. E. Pereira, F. R. Wagner, E. T. Silva Jr., F. C. Carvalho. “DERAF: A High-Level Aspects Framework for Distributed Embedded Real-Time Systems Design”. *In: Proc. of 10th International Workshop on Early Aspects* , Springer, 2007, pp. 55-74.

[7] Wehrmeister, M.A., Freitas, E.P., Pereira, C.E., Wagner, F.R. “Applying Aspect-Orientation Concepts in the Model-Driven Design of Distributed Embedded Real-Time Systems”. *In: Proc. of 10th IEEE International Symposium on Object/component/serrvice-oriented Real-time Distributed Computing (ISORC'07), Springer* , 2007, pp. 221-230.

[8] A. Tesanovic, et al. “Aspects and Components in Real-Time System Development: Towards Reconfigurable and Reusable Sofftware”, *Journal of Embedded Computing*, IOS Press, v.1, n.1, 2005.

[9] E. P. Freitas, P. Söderstam, W. O. Morais, C. E . Pereira, T. Larsson. “Adaptable Middleware for Heterogeneous Wireless Sensor Networks”, *In Proc. 10 th European Agent Systems Summer School (EASSS08),* 2008. pp.17-24.

[10] IEEE Std 716-1995, 1995. IEEE standard test language for all systems-Common/Abbreviated Test Language for All Systems (C/ATLAS), IEEE, Inc.

[11] K. Romer, O. Kasten, and F. Mattern, “Middleware challenges for wireless sensor networks,” *ACM SIGMOBILE Mobile Communication and Communications Review* , vol. 6, no. 2, 2002.

[12] I. F. Akyildiz and W. Su and Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey”, IEEE Computer, vol. 38, no. 4, pages 393-422, Mar. 2002.

[13] Object Management Group (OMG). Distribution Service for Real-time Systems (DSS) Specification. Version 1.2. January 2007.

[14] W. Heinzelman, A. Murphy, H. Carvalho and M. Perillo,"Middleware to Support Sensor Network Applications," *IEEE Network Magazine Special Issue*. Jan. 2004.

[15] J. King, R. Bose, Hen-I Yang; S. Pickles, A. Helal. Atlas: A Service-Oriented Sensor Platform: Hardware and Middleware to Enable Programmable Pervasive Spaces. In: *Proc of 31st IEEE Conference on Local Computer Networks* , 2006. pp. 630 - 638.

[16] OSGi Alliance. OSGi Service Platform, Core Specification, Release 4, Version 4.1. May 2007.

[17] R. E. Schantz, J. P. Loyall, D. C. Schmidt, C. Rodrigues, Y. Krishnamurthy, and I. Pyarali. “Flexible and Adaptive QoS Control for Distributed Real-time and Embedded Middleware”, *In Proc. of 4th IFIP/ACM/USENIX International Conference on Distributed Systems Platforms* , *Springer* , 2003.

[18] Object Management Group (OMG). Common Object Request Broker Architecture: Core Specification. Version 3.0.3. March 2004.

[19] R. Vanegas, J. Zinky, J. Loyall, D. Karr, R. Schantz, and D. Bakken, “QuO's Runtime Support for Quality of Service in Distributed Objects", *In Proc. of Middleware 98* , the IFIP International Conference on Distributed Systems Platform and Open Distributed Processing, Sept 1998.

[20] Heimfarth, T., Janacik, P. Cross-layer Architecture of a Distributed OS for Ad Hoc Networks. In: Proceedings of the International Conference on Autonomic and Autonomous Systems, 2006. ICAS '06. pp. 52- 52, 2006,

[21] Heimfarth, T., Janacik, P., Rammig, F. J. Self-Organizing Resource-Aware Clustering for Ad Hoc Networks. In: Proceedings of the 5th IFIP Workshop on Software Technologies for Future Embedded & Ubiquitous Systems (SEUS 2007), Santorini Island, Greece, Mai 2007

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 601–608

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Embedded Control Systems Design based on RT-DEVS and Temporal Analysis using UPPAAL

Angelo Furfaro and Libero Nigro
Laboratorio di Ingegneria del Software
Dipartimento di Elettronica Informatica e Sistemistica
Universitá della Calabria, 87036 Rende (CS) Italy
http://www.lis.deis.unical.it
Email: a.furfaro@deis.unical.it, l.nigro@unical.it

***Abstract*—This work is concerned with modelling, analysis and implementation of embedded control systems using RT-DEVS, i.e. a specialization of classic DEVS (Discrete Event System Specification) for real-time. RT-DEVS favours model continuity, i.e. the possibility of using the same model for property analysis (by simulation or model checking) and for real time execution. Special case tools are proposed in the literature for RT-DEVS model analysis and design. In this work, temporal analysis exploits an efficient translation in UPPAAL timed automata. The paper shows an embedded control system model and its exhaustive verification. For large models a simulator was realized in Java which directly stems from RT-DEVS operational semantics. The same concerns are at the basis of a real-time executive. The paper discusses the implementation status and, finally, indicates research directions which deserve further work.**

***Index Terms*—DEVS, real-time constraints, embedded control systems, model continuity, temporal analysis, timed automata, model checking, Java.**

## I. INTRODUCTION

THERE is a general agreement today about the importance of using formal tools for rigorous development of real-time systems which in general have safety and time critical requirements to fulfil. However, a known hard problem for the developer is how to ensure that a given formal model of a system, preliminarily analyzed from both functional and temporal viewpoints, is correctly reproduced in an implementation. This paper describes some work aimed to the realization of tools for modelling, analysis and implementation of embedded control systems, specifically for experimenting with model continuity [1], [2], i.e. seamless development where the same model is used both for property analysis (through simulation or model checking) and for real time execution. The modelling language is RT-DEVS [3], [4], i.e. is a specialization of classic DEVS (Discrete Event System Specification) [5] with a weak synchronous communication model and constructs for expressing timing constraints. RT-DEVS owes to DEVS for both atomic and coupled component formalization and model continuity. Special case tools are reported in the literature [4] to support a development methodology for RT-DEVS.

The original contribution of this work is twofold:

- proposing a mapping of the fundamental phases of modelling and safety/temporal analysis of RT-DEVS systems in terms of the popular and efficient UPPAAL toolbox with timed automata [6], [7], [8]
- building concrete tools in Java for RT-DEVS simulation and final system implementation. The Java-based approach aims to improve applicability and portability of RT-DEVS software.

This paper introduces RT-DEVS and its operational semantics, then a transformation process of RT-DEVS specifications into UPPAAL is suggested for exhaustive verification activities based on model checking. The approach is demonstrated through a realistic embedded control system. After that, current implementation status of Java-based development tools and programming style are clarified. Prototype tools were achieved by adapting existing tools for ActorDEVS [9], [10]. Finally, conclusions are presented with an indication of directions of further work.

## II. RT-DEVS DEFINITIONS

### A. DEVS Basics

DEVS [5] is a widespread modelling formalism for concurrent and timed systems, founded on systems theory concepts. A DEVS system consists of a collection of one or more components. Two types of components exist: *atomic* (or behavioural), and *coupled* (or structural) components. A DEVS atomic component is a tuple $AM$ defined as $AM =< X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta >$ where:

- $X$ is the set of input values
- $S$ is a set of states
- $Y$ is the set of output values
- $\delta_{int} : S \rightarrow S$ is the *internal transition* function
- $\delta_{ext} : Q \times X \rightarrow S$ is the *external transition* function, where $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$ is the set of *total states*, $e$ is the *elapsed time* since last transition
- $\lambda : S \rightarrow Y$ is the *output function*
- $ta : S \rightarrow \mathbb{R}^{+}_{[0,\infty]}$ is the time *advance function*.

The sets $X$, $S$ and $Y$ are typically products of other sets. $S$, in particular, is normally the product of a set of *control states* (also said *phases*) and other sets built over the values of a certain number of variables used to describe the component at hand. Informal semantics of above definitions are as follows. At any time the component is in some state $s \in S$. The component can remain in s for the time duration (*dwell-time*) $ta(s)$. $ta(s)$ can be 0, in which case $s$ is said a transitory state, or it can be $\infty$, in which case it is said a passive state because the component can remain forever in $s$ if no external event interrupts. Provided no external event arrives, at the end of (supposed finite) time value $ta(s)$, the component moves to its next state $s' = \delta_{int}(s)$ determined by the internal transition function $\delta_{int}$. In addition, just *before* making the internal transition, the component produces the output computed by the output function $\lambda(s)$. During its stay in $s$, the component can receive an external event $x$ which can cause $s$ to be exited earlier than $ta(s)$. Let $e \leq ta(s)$ be the elapsed time since the entering time in $s$. The component then exits state $s$ moving to next state $s' = \delta_{ext}(s, e, x)$ determined by the external transition function $\delta_{ext}$. As a particular case, the external event $x$ can arrive when $e = ta(s)$. In this (*collision*) case two events occur simultaneously: the internal transition event and the external transition event. A collision resolution rule is responsible for ranking the two events and determining the next state. After entering state $s'$, the new time advance value $ta(s')$ is computed and the same story continues. It should be noted that there is no way to directly generate an output from an external transition. To achieve this effect a transitory phase, used as destination of the external transition and whose lambda function generates the desired output, can be introduced (see Fig. 4).

In practice, an atomic component receives its inputs from typed *input ports* and similarly, generates outputs through typed *output ports*. Actually $X$ is a set of pairs $< inp, v >$ where $inp$ is an input port and $v$ the type of values which can flow through $inp$. $Y$ is a set of pairs $< outp, v >$ where $outp$ is an output port. Ports are architectural elements which enable modular system design. A component refers only to its interface ports. It has no knowledge about the identity of cooperating partners. A coupled component (subnet) is an interconnection of existing atomic or coupled (hierarchical) components. Formally, it is a structure $CM$ defined as $CM = (X, Y, D, \{M_d | d \in D\}, EIC, EOC, IC)$, where:

- $X$ and Y are input and output sets of the coupled component
- $D$ is a set of (sub) component identifiers (or names)
- $M$ is a set of (sub) DEVS components whose interconnection gives rise to the coupled model
- $EIC$ is the external to internal coupling function (for routing external events to internal components)
- $EOC$ is the internal to external coupling function (for routing internally generated events to the external environment of the coupled component)
- $IC$ is the internal to internal coupling function.

### B. RT-DEVS Concepts

RT-DEVS [4] refines basic DEVS with the following concepts.

1) The dwell-time $ta(s)$ in a state now mirrors the execution time of an *activity* associated with the state. In particular, the execution time is specified by a (dense and static) time interval $[lb, ub]$, where lower and upper bounds $lb, ub \in \mathbb{R}^+_{[0,\infty]}$, $0 \leq lb \leq ub$, express uncertainty in the activity duration. Default interval of passive states is $[\infty, \infty]$ and can be omitted. Transitory (or immediate) states have interval $[0, 0]$.
2) Non determinism is assumed as collision resolution rule.
3) The communication model is weak synchronous, i.e. non blocking with (possible) message loss. At any communication, an output event is always immediately consumed. If the receiver is not ready, the message is lost. If both sender and receiver are ready to communicate, the output event is converted into an input event which is instantly received.

A time interval $[lb, ub]$ is made absolute at the instant in time $\tau$ the corresponding state $s$ is entered. An internal transition outgoing $s$ can occur at any time greater than or equal $\tau + lb$ but, to avoid a timing violation, before or at $\tau + ub$. An external transition fires upon synchronization on an input event independently of the dwell-time of current phase. It is assumed that a self-loop external transition does not restart timing in current phase. Pre-emption and restarting of current timing, when desired, can be simulated with the help of an transitory phase. Graphically (see e.g. Fig. 2), an internal transition is depicted by a thin oriented edge terminating with a dashed arrow which specifies the execution of the lambda (output) function, which can be void. An external transition is instead drawn by a thick oriented edge. Sending event $ev$ through outport $OP$ is denoted by the syntax $OP!ev$. Similarly, readiness to accept event $ev$ through input port $IP$ is expressed by $IP?ev$. The abstract executor of RT-DEVS initializes current time to $0$ and iterates the following two basic steps.

1) The next minimal time at which new internal transitions can fire is determined and become the current time.
2) All candidate internal transitions which can occur at current time are determined. Let $C_i$ be an atomic component with one such a transition. Let the lambda function of current state of $C_i$ consist of $OP!ev$. Let $C_j$ be a component coupled with $C_i$ where input port $IP$ matches output port $OP$ of $C_i$. Provided $C_j$ has an outgoing transition from current state annotated with $IP?ev$, the two transitions (internal in $C_i$ and external in $C_j$) are immediately executed with the event sent by $C_i$ synchronously transmitted to $C_j$. In the case $C_j$ is not ready to receive $C_i$ event, the output transition in $C_i$ is still made but the event gets lost. The above activity is repeated for each candidate internal transition. When the candidate set empties, the executor goes back to step 1.

It is worthy of note that while weak synchronization is a useful feature in general real time systems (e.g., a message with a sensor reading can be lost for a missing synchronization, in which case a controller can use previous sensor data), it increases the burden of the RT-DEVS modeller when the system cannot tolerate synchronization losses. Model validation through simulation or verification can help in assessing correct system behaviour.

## III. A Traffic Light Controller

The following describes the modelling of a Traffic Light Control system (TLC) [11]. In the proposed scenario, the traffic flow at an intersection between an avenue and a street is regulated by two traffic lights. The lights are operated by a control device (controller) that, in normal conditions, alternates in a periodic way the traffic flow in the two directions. In addition, the controller is able to detect the arrival of an ambulance and to handle this exceptional situation by allowing the ambulance crossing as soon as possible and in a safe way. For the sake of simplicity, it is assumed that at most one ambulance can be in the closeness of the intersection at a given time. During normal operation conditions, the sequence green-yellow-red is alternated on the two directions with the light held green for $45$ time units (tu), yellow for $5$ tu and red on both directions for 1 tu. The intersection is equipped with sensors able to detect the presence of an ambulance at three different positions during its crossing. As soon as the ambulance arrival is detected, a signal named "approaching" is sent to the controller. When the ambulance reaches the nearness of the intersection the signal "before" is issued. After the ambulance completes the crossing the signal "after" is generated. The controller reacts to the "approaching" event by leading the intersection to a safe state, i.e. bringing both lights on red.

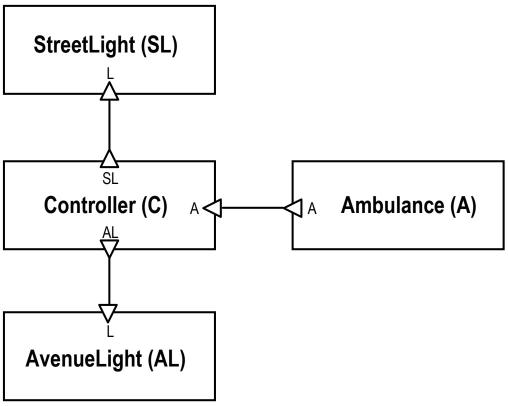


Fig. 1. Traffic light coupled model

When the signal "before" is received, the controller switches to green the light on the ambulance's arrival direction. After the ambulance leaves the intersection ("after" event) the controller turns the green light to red and resumes its normal sequence. Fig. 1 illustrates an RT-DEVS coupled model of the TLC system which is made of four connected components: there are two instances of the Light component, which respectively correspond to the light on the avenue and that on the street, one Ambulance component, which models the behaviour of the sensing equipments of the intersection, and one Controller component which implements the above described control logic. Couplings in Fig. 1 are realized between matching input/output ports. $X/Y$ sets for the Controller are as follows:

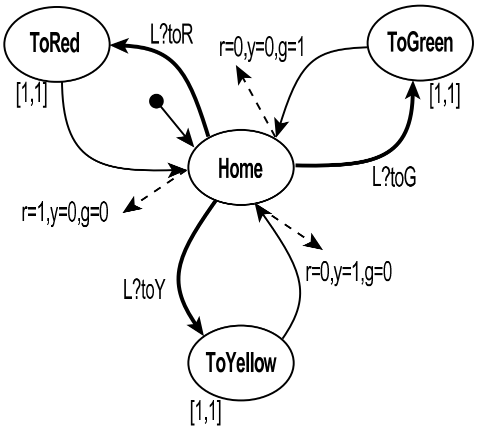


Fig. 2. Light behaviour

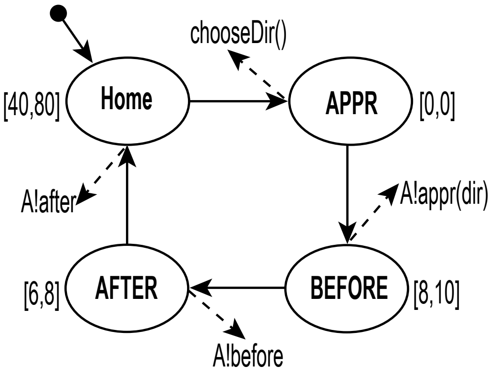


Fig. 3. Ambulance behaviour

```
X={<A,appr>,<A,before>,<A,after>}
Y={<SL,toR>,<SL,toY>,<SL,toG>,<AL,toR>,
    <AL,toY>,<AL,toG>}
```

Component behaviour is specified in Figg. from 2 to 4 where an oval box represents a phase of the component. The complete state set $S$ obviously depends also on the component local variables. For instance, the Controller has a dir variable whose value indicate the avenue or the street, and logical variable amb where information about an arriving ambulance is maintained when current phase of the controller cannot be pre-empted. Similarly, light components keep the light status in the three logical variables r,y, and g. A light component (Fig. 2) is normally in the Home phase with default interval $[\infty, \infty]$. The arrival of a toR, toY or toG event causes an external transition respectively to toRed, toYellow or toGreen phase which is then exited after 1 time unit by an internal transition reaching again Home. The lambda function associated with the internal transitions specifies the required state changes in the light.

Behaviour of the ambulance (Fig. 4) is cyclic. After a non deterministic time in $[40, 80]$, the ambulance announces itself by choosing an arriving direction and sending the appr event to the controller. From the BEFORE phase and after a time in $[8, 10]$ the ambulance sends a before event to the controller. Finally, form the AFTER phase the ambulance signals its passage through the intersection by sending an after event with an elapsed time in $[6, 8]$. In Fig. 4 the normal and exceptional

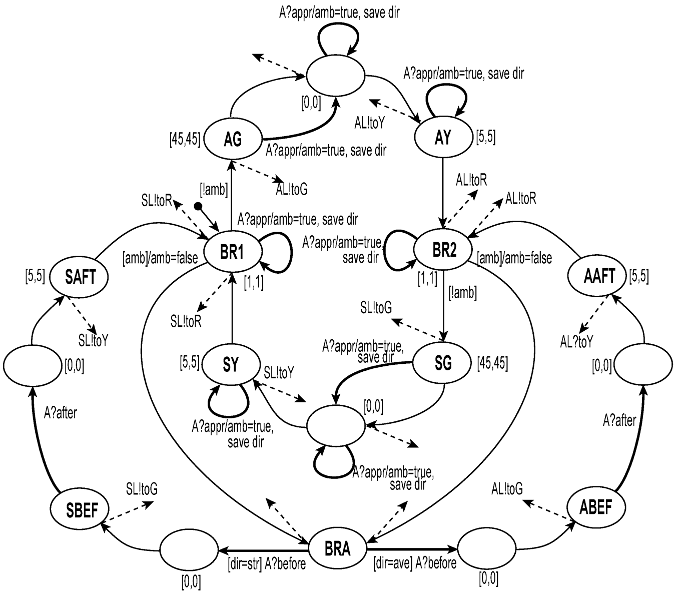


Fig. 4. Controller behaviour

behaviours of the controller can be distinguished. The initial phase is BR1 (both lights reds). Under normal behaviour, the controller steps through a light cycle (e.g. from BR1, to AV to AY to BR2 for the avenue, and from BR2 to SG, to SY to BR1 for the street). It should be noted that a "both reds" condition (BR1 or BR2) is always maintained for 1 time unit. Avenue and street cycles strictly alternate. A normal cycle is started provided no ambulance is sensed. During a light cycle the arrival of ambulance pre-empts normal behaviour. In particular, a green phase (AG or SG) is immediately abandoned by anticipating the next yellow phase and then finishing the cycle. However, current yellow phase is never pre-empted. All of this guarantees the duration of the yellow phase (in the example in [11] it was erroneously made possible, in worst case conditions, that a yellow phase doubles its duration). It should be noted the efforts taken in Fig. 4 for not losing the approaching signals. As soon as an ambulance is sensed, the logical variable amb is set to true. At cycle end, the presence of an ambulance requires an exceptional behaviour to be executed by first reaching the BRA (both reds with ambulance) phase. From BRA, and depending on the arriving direction of the ambulance, the controller senses events from the ambulance and commands accordingly the light by turning it first green, then yellow after ambulance passage and finally red. Ambulance events (e.g. before and after) are processed by external transitions. Light control is instead realized through internal transitions. Following an exceptional behaviour, the controller restarts the normal cycle by giving the turn to the other direction.

### *A. Property Requirements for the TLC*

The TLC has safety and (bounded) liveness (e.g. deadline) properties, besides absence of deadlock or livelock conditions.

1) *Traffic must never be allowed in both directions simultaneously*. For safety reasons it is required that the status of the traffic lights be consistent at all times. To avoid accidents among vehicles crossing the intersection, when on a direction the light is green or yellow, thus allowing traffic in the direction, the light on the opposite direction must be red.
2) *Lights should be both reds at a before event*. No vehicle should be allowed to cross the intersection at a before event.
3) *Deadline of* $3$ *tu for turning green the light after a before event*. Assuming that it takes at least $4$ tu for the ambulance to reach the intersection from the time instant of the before signal, it follows that there exists a deadline of 3 tu for the controller to turn green the light on the arriving direction, also considering that a light takes 1 tu for changing its status.
4) *Correct sequencing of the lights on each direction*. A correct behaviour requires that only transitions from red to green, from green to yellow and from yellow to red should be allowed. A transition out of this sequence denotes a wrong sequence.
5) *The ambulance must be live*. In particular, after signalling an approach, it must be guaranteed that the ambulance model comes back to its Home phase.

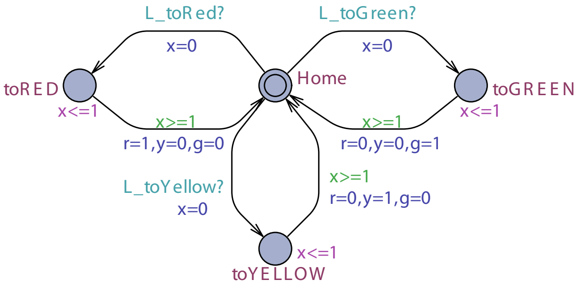


Fig. 5. Light template

## IV. Temporal Analysis using Uppaal

Weak synchronization and message losses increase the need for functional, safety and temporal analysis of an RT-DEVS model. In this work an RT-DEVS model is preliminary transformed into Uppaal [6] for model checking. Uppaal was chosen because it supports data variables and weak synchronization through broadcast channels [12]. The following summarizes the translation rules.

- An RT-DEVS component is mapped onto an Uppaal template, which has a local clock $x$.
- Phases of the source component correspond one-to-one to locations of the template.
- Each pair of matching ports (e.g. the output port A of Ambulance and the input port A of Controller) together with a data/control symbol, is mapped on to a broadcast channel. For instance, broadcast channels A_appr, A_before and A_after are shared between Ambulance and Controller etc.
- Templates receive as parameters the broadcast channels corresponding to used input/output ports.

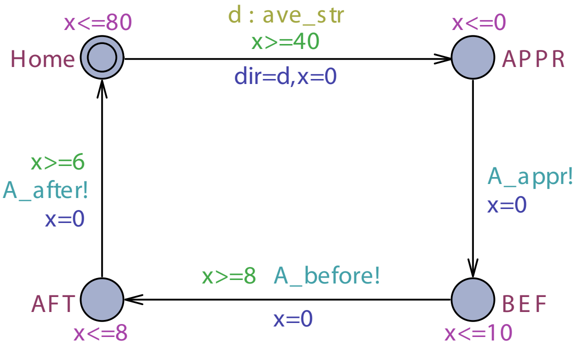


Fig. 6. Ambulance template

TABLE I
UPPAAL QUERIES FOR PROPERTY ANALISYS OF TLC

| Property | Query | Result |
|---|---|---|
| Absence of deadlocks | A[] !deadlock | satisfied |
| Lights consistency | A[] (AL.g==1\|\|AL.y==1) imply SL.r==1 | satisfied |
| Lights consistency | A[] (SL.g==1\|\|SL.y==1) imply AL.r==1 | satisfied |
| Lights must never be both green | E<> SL.g==1 && AL.g==1 | not satisfied |
| Ambulance is live | A.APPR --> A.Home | satisfied |
| Deadline checking | A[] flag imply z<=3 | satisfied |

- Shared communication data, e.g. the dir variable used by Ambulance and Controller, become global declarations.
- A strict time interval $[lb, ub]$ of a phase PH of an RT-DEVS component implies the invariant $x \leq ub$ is added to location PH. Default time interval $[\infty, \infty]$ is implicit. Time interval $[0, 0]$ of a transitory phase is mapped on the invariant $x \leq 0$. UPPAAL requires bounds of a time interval to be expressed by naturals.
- An internal transition of the RT-DEVS model is associated with a timed edge having the guard $x \geq lb$. The update portion of the command on the edge contains the effect of the output function. An external transition is associated with an untimed edge which in turn relates to an input synchronization with a broadcast channel.

The above rules were applied to obtain the models in Figg. from 5 to 7 which depict the UPPAAL version of RT-DEVS TLC components. In Fig. 6, random choice of the ambulance arriving direction is simply achieved by non-deterministic selection, on the edge between Home and APPR locations, of the value of the local variable d between ave and str values (type ave_str is an alias of int[ave,str]). As one can see, the UPPAAL templates correspond as close as possible to source RT-DEVS components. Therefore, the translation can be easily automated. The resultant UPPAAL system model is the parallel composition of one instance of the Controller template, two instances of the Light template and one instance of the Ambulance.

### A. Verification of the TLC

The timed automata model of the TLC was verified using the UPPAAL version 4.1.0. Table I illustrates some TCTL queries issued to the UPPAAL verifier used for property analysis.

Absence of deadlocks confirms the TLC model correctly behave despite weak synchronization and (possibly) message loss. That the unsafe state of both lights green is never reached is checked by asking the verifier if there exists a state in the state graph where the g data of both lights is 1. In addition, it was verified that whenever the traffic is allowed in one direction (the light is green or yellow), the light is red on the other direction.

Correct sequencing of lights was verified by introducing three additional variables in the Light template for storing the previous status of the light, by changing the Light behaviour so as to conserve previous status at any new assignment, and by checking that it is always true that a green status is preceded by the red status etc. These details and queries are omitted for simplicity.

A few additional words relate to deadline checking. The UPPAAL model was decorated by introducing the global logical variable flag and the extra clock $z$. Variable flag is set to true in the Ambulance template when the before event is sent to controller, and reset in the Light template (therefore in both instances of the template) when the green status is installed (on the exiting edge from the toGreen location in Fig. 5). It was found that not only the required deadline is fulfilled but that in reality 1 tu is always sufficient for the controller, following a before signal, to turn green the light in the arriving direction of the ambulance.

## V. IMPLEMENTATION STATUS

RT-DEVS was prototyped in Java using an adaptation of the ActorDEVS lean agent-based framework [9], [10]. The following provides some implementation hints and gives a flavour of the programming style. Both discrete and dense time models are supported, through the class hierarchy (interfaces are underlined): `Time`, `AbsoluteTime`, `RelativeTime`, `TimeInterval`, `AbsoluteDiscreteTime`, `AbsoluteDenseTime`, `RelativeDiscreteTime`, `RelativeDenseTime`, `DiscreteTimeInterval`, `DenseTimeInterval`. A concrete time object has a `value()` method which returns a `long` for discrete time, and a `double` for dense time. An RT-DEVS atomic component must be programmed as a class which derives directly or indirectly from the `RTDEVS` abstract base class, which provides the contract of operations (see the extract in Fig. 8) and basic behaviour.

A specific component must implement the abstract methods of `RTDEVS` in order to specify its specific behaviour. For simulation purposes the `activity()` method can be left to its default (no-operation) body. Phases are coded as integers. Internal and external transitions return the int of the next phase. It should be noted that component methods have direct access to the whole state by accessing the component local data variables. The `ti()` method returns the (dense or discrete) time interval associated with the given state. Method `now()` returns the `AbsoluteTime` value of current time.

Typed input/output ports are supported respectively by parametric classes `Input<V>` and `Output<V>`. Typically, `V` is a

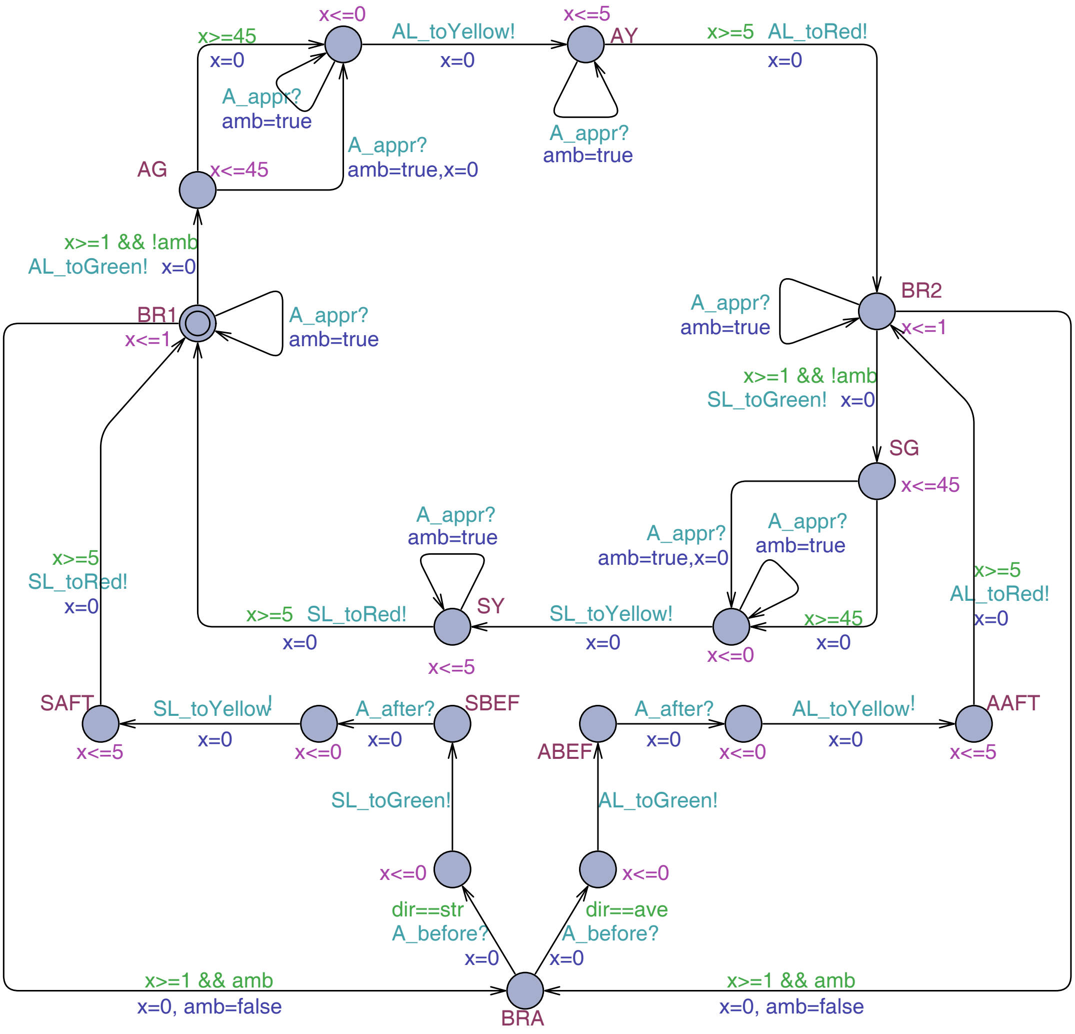


Fig. 7. Controller template

```
public abstract int delta_int( int phase );
public abstract int delta_ext( int phase, RelativeTime e, Message x );
public abstract void lambda( int phase );
public abstract RelativeTime ta( int phase );
public abstract TimeInterval ti( int phase );
public void activity( int phase ){}
public AbsoluteTime now();
```

Fig. 8. An extract of RTDEVS atomic components programming interface

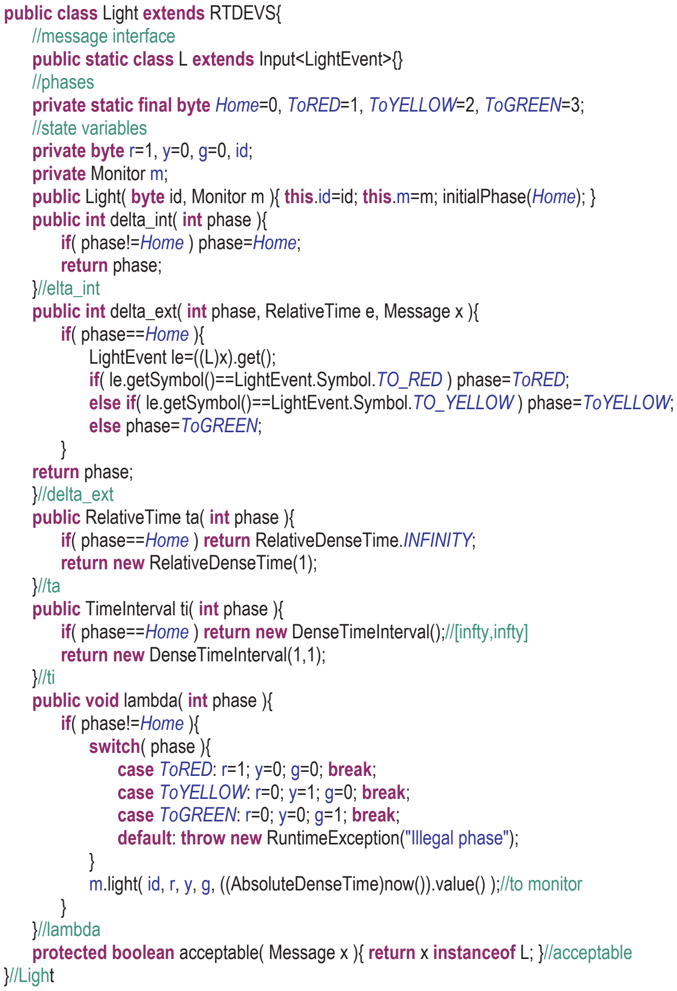
```
public class LightEvent {
    public static enum Symbol{ TO_RED, TO_YELLOW, TO_GREEN };
    private Symbol symbol;
    public Symbol getSymbol(){ return symbol; }
    public void setSymbol( Symbol symbol ){ this.symbol=symbol; }
}//LightEvent
```

Fig. 9. Class of light events

user defined class which specifies the data/control symbols which can flow through the port. `Input` is a subclass of `Output`. Each component exports its input port types. Output ports are created by a configurer (e.g. the `main()` method) and passed to relevant components e.g. at construction time. The configurer is also in charge of linking matching ports for establishing a coupled model. Programming style is exemplified by showing details of the Light atomic component. Light events were modelled as instances of the `LightEvent` class (Fig. 9). The Light component, shaped for prototyping and simulation purposes, is illustrated in Fig. 10.

For components with non punctual time intervals (e.g. Ambulance and Controller) the `ta()` method returns a number uniformly distributed in the time interval of current phase.

Java TLC model was executed using dense time and the `RTDEVS_Simulation` control engine which mimics the RT-DEVS operational semantics. `RTDEVS_Simulation` receives the (`AbsoluteDenseTime`) simulation time limit (e.g. $10^7$) and a simulation clock (here a `SimulationDenseTimeClock`). `RTDEVS` maintains a priority queue of timers ranked by ascending fire times (absolutized `ta` values). The engine fires most imminent



```
public class Light extends RTDEVS{
    //message interface
    public static class L extends Input<LightEvent>{}
    //phases
    private static final byte Home=0, ToRED=1, ToYELLOW=2, ToGREEN=3;
    //state variables
    private byte r=1, y=0, g=0, id;
    private Monitor m;
    public Light( byte id, Monitor m ){ this.id=id; this.m=m; initialPhase(Home); }
    public int delta_int( int phase ){
        if( phase!=Home ) phase=Home;
        return phase;
    }//elta_int
    public int delta_ext( int phase, RelativeTime e, Message x ){
        if( phase==Home ){
            LightEvent le=((L)x).get();
            if( le.getSymbol()==LightEvent.Symbol.TO_RED ) phase=ToRED;
            else if( le.getSymbol()==LightEvent.Symbol.TO_YELLOW ) phase=ToYELLOW;
            else phase=ToGREEN;
        }
    return phase;
    }//delta_ext
    public RelativeTime ta( int phase ){
        if( phase==Home ) return RelativeDenseTime.INFINITY;
        return new RelativeDenseTime(1);
    }//ta
    public TimeInterval ti( int phase ){
        if( phase==Home ) return new DenseTimeInterval();//[infty,infty]
        return new DenseTimeInterval(1,1);
    }//ti
    public void lambda( int phase ){
        if( phase!=Home ){
            switch( phase ){
                case ToRED: r=1; y=0; g=0; break;
                case ToYELLOW: r=0; y=1; g=0; break;
                case ToGREEN: r=0; y=0; g=1; break;
                default: throw new RuntimeException("Illegal phase");
            }
            m.light( id, r, y, g, ((AbsoluteDenseTime)now()).value() );//to monitor
        }
    }//lambda
    protected boolean acceptable( Message x ){ return x instanceof L; }//acceptable
}//Light
```

Fig. 10. Class Light of the TLC

internal transitions one at a time and updates the simulation clock to the fire time accordingly. The output function then sends synchronously its message to the coupled component. In the case the partner component is not ready for synchronization, the sent message is simply lost. During simulation, a `Monitor` object (transducer) gets informed of event occurrences and checks system properties (e.g. it counts the number of times the bad state green-green of the two lights is reached, and measures the maximal time distance between the occurrence time of the green light in the arriving direction of the ambulance, and that of the immediately preceding before event, etc.). Also under simulation, the TLC was found to be temporally correct.

For real-time execution, RT-DEVS naturally requires a multi-processor implementation (each component runs on its own processor, as was assumed by temporal analysis). The `ta()` function is no longer useful. The `activity()` method should be programmed with the (sub)algorithms to be carried out in each phase of the component. All other methods remain unchanged. Of course, a real-time executive has to possibly compensate for violations of activity durations. An activity can terminate earlier than its lower bound duration or after its upper bound. In the first case the engine can delay the firing of the internal transition until the real time clock reaches the lower bound. In the latter case activity interruption and concepts of adaptive scheduling and imprecise computation [13] could help. As a particular scenario, an RT-DEVS model could be analyzed and executed on a single processor, by ensuring atomicity and mutual exclusion of activities.

## VI. Conclusion

This paper reports about specification, analysis and Java implementation of RT-DEVS systems operated under model continuity. Model checking is enabled by a translation onto timed automata of UPPAAL. For large models an achieved discrete-event simulation tool can be exploited. Java implementations rely on a minimal, efficient and customizable agent framework [9], [10].

On-going and future work is directed at:

- experimenting with real-time executives using the Real-time Specification for Java platform [14]
- extending the approach to the distributed context using standard middleware like HLA/RTI or real-time CORBA
- building development tools for visual modelling, prototyping/simulation, and automatic generation of Java code and UPPAAL XML code.

## References

[1] X. Hu and B. Zeigler, “Model continuity to support software development for distributed robotic systems: A team formation example,” *Journal of Intelligent and Robotic Systems*, vol. 39, no. 1, pp. 71–87, 2004.

[2] ——, “Model continuity in the design of dynamic distributed real-time systems,” *IEEE Trans. Syst., Man, Cybern. A*, vol. 35, no. 6, pp. 867–878, 2005.

[3] J. Hong, H. Song, T. Kim, and K. Park, “A real-time discrete-event system specification formalism for seamless real-time software development,” *Discrete Event Systems: Theory and Applications*, vol. 7, pp. 355–375, 1997.

[4] H. Song and T. Kim, “Application of real-time DEVS to analysis of safety-critical embedded control systems: railroad-crossing example,” *Simulation*, vol. 81, no. 2, pp. 119–136, 2005.

[5] B. P. Zeigler, H. Praehofer, and T. Kim, *Theory of modeling and simulation*, 2nd ed. New York: Academic Press., 2000.

[6] G. Behrmann, A. David, and K. G. Larsen, “A tutorial on UPPAAL,” in *Formal Methods for the Design of Real-Time Systems*, ser. LNCS 3185, M. Bernardo and F. Corradini, Eds. Springer, 2004, pp. 200–236.

[7] F. Cicirelli, A. Furfaro, and L. Nigro, “Using TPN/Designer and UPPAAL for modular modelling and analysis of time-critical systems,” *International Journal of Simulation Systems, Science & Technology*, vol. 8, no. 4, pp. 8–20, 2007, special Issue on Frameworks and Applications in Science and Engineering.

[8] A. Furfaro and L. Nigro, “Modelling and schedulability analysis of real-time sequence patterns using time Petri nets and UPPAAL,” in *Proc. of International Workshop on Real Time Software (RTS’07)*, October 16 2007, pp. 821–835.

[9] F. Cicirelli, A. Furfaro, and L. Nigro, “A DEVS M&S framework based on Java and actors,” in *Proc. of 2nd European Modeling and Simulation Symposium (EMSS’06)*, Barcelona, Spain, October 4-6 2006.

[10] ——, “Actor-based simulation of PDEVS systems over HLA,” in *Proc. 41st Annual Simulation Symposium (ANSS’08)*, 2008, pp. 229–236.

[11] S. C. V. Raju and A. C. Shaw, “A prototyping environment for specifying and checking Communicating Real-time State Machines,” *Software–Practice and Experience*, vol. 24, no. 2, pp. 175–195, 1994.

[12] Uppaal. [Online]. Available: http://www.uppaal.com

[13] W. A. Halang, “Load adaptive dynamic scheduling of tasks with hard deadlines useful for industrial applications,” *Computing*, vol. 47, pp. 199–213, 1992.

[14] RTSJ. [Online]. Available: http://jcp.org/aboutJava/communityprocess/first/jsr001/rtj.pdf

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 609 – 615

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Study of different load dependencies among shared redundant systems

Ján Galdun *, **
* Laboratoire GIPSA-Lab (GIPSA-Lab UMR 5216 CNRS-INPG-UJF) BP 46, F-38402 Saint Martin d'Hères Cedex, France
Email: Jan.Galdun@tuke.sk

Jean-Marc Thiriet
Laboratoire GIPSA-Lab (GIPSA-Lab UMR 5216 CNRS-INPG-UJF) BP 46, F-38402 Saint Martin d'Hères Cedex , France
Email: Jean-Marc.Thiriet@ujf-grenoble.fr

Ján Liguš
* * Department of Cybernetics and Artificial Intelligence, Technical University of Košice,
Letná 9, 04012 Košice, Slovakia
Email: Jan.Ligus@tuke.sk

*Abstract*—**The paper presents features and implementation of a shared redundant approach to increase the reliability of networked control systems. Common approaches based on redundant components in control system use passive or active redundancy. We deal with quasi-redundant subsystems (shared redundancy) whereas basic features are introduced in the paper. This type of redundancy offers several important advantages such as minimizing the number of components as well as increasing the reliability. The example of a four-rotor mini-helicopter is presented in order to show reliability improving without using any additional redundant components. The main aim of this paper is to show the influence of the load increasing following different scenarios. The results could help to determine the applications where quasi-redundant subsystems are a good solution to remain in a significant reliability level even if critical failure appears.**

**Keywords: Shared redundancy, Dependability, Networked control systems**

## I. Introduction

To BE able to obtain relevant results of reliability evaluations for complex systems, it is necessary to describe the maximum of specific dependencies within the studied system and their influences on the system reliability. Different methods or approaches for control systems' reliability improvement are developed in order to be applied to specific subsystems or to deal with dependencies among subsystems. A classical technique consists in designing a fault-tolerant control [12] where the main aim is to propose a robust control algorithm. Guenab and others in [4] deal with this approach and reconfiguration strategy in complex systems, too.

On the other side is the design of reliable control architectures. Probably the most used technique is to consider the redundant components which enlarge the system structure and its complexity too. Active and passive redundancy is the simplest way how to improve dependability attributes of the systems such as reliability, maintainability, availability, etc [8]. However, as it was mentioned the control structure turns to be more complex due to an increasing number of components as well as the number of possible dependencies among components.

The paper introduces complex networked control architecture based on cascade control structure. The cascade structure was chosen purposely due to its advantages. This structure is widely used in industrial applications thanks to positive results for quality of control which are already described and generally known [2]. On the other side it offers some possibilities of system reliability improvement. There are potentially redundant components such as controllers (primary, secondary). If more than one network is implemented we could consider them as potentially redundant subsystems too. Finally if the physical system allows it, it is possible to take profit from sensors. The cascade structure and other features are introduced in more details in the third part.

The paper is organised as follows. After bringing closer the research background, the shared redundancy is introduced. The controllers and networks are presented in more details in order to show some dependencies which could be appeared when a shared redundancy approach is implemented. In the next part are presented networked topologies considered as cascade control (CC) structure of the 4-rotor mini-helicopter (drone) model [3]. Using Petri nets were prepared the models of the introduced quasi-redundant components as well as drone's control structure. A simple model of the two quasi-redundant subsystems is evaluated. Finally, are proposed the simulation results of the mentioned simple two components model as well as the model of the complex drone's structure with short conclusion.

## II. Research Background

Control architecture design approach was taken into account by Wysocki, Debouk and Nouri [13]. They present shared redundancy as parts of systems (subsystems) which could replace another subsystem in case of its failure. This feature is conditioned with the same or similar function of the subsystem. Wysocki et al. introduce the shared redundant architecture in four different examples illustrated on "X-by-Wire" systems used in automotive applications. Presented results shown advantages of this approach in control architecture design.

The shared redundancy approach involves the problematic of a *Load Sharing* [1]. Thus, some of the components take part of the load of the failed components in order to let the system in functional mode. Consideration of the load sharing

978-83-60810-14-9/08/$25.00 © 2008 IEEE

in mechanical components is presented by Pozsgai and others in [11]. Pozsgai and others analyze this type of systems and offer mathematical formalism for simple system 1-out-of-2 and 1-out-of-3. Also there are some mathematical studies [1] of several phenomena appeared on this field of research. Bebbington and others in [1] analyze several parameters of systems such as survival probability of load shared subsystems.

## III. Shared Redundancy

Specific kind of redundant subsystems which have similar features such as active redundancy however gives us some additional advantages which will be introduced in further text. This kind of spares represents another type of redundant components which are not primary determined as redundant but they are able to replace some other subsystem if it is urgently required. This type of redundancy is referred as *shared redundancy* [13] or *quasi-redundancy* [6]. Due to its important advantages it is useful to describe this kind of spares in order to show several non-considered and non-evaluated dependencies which could have an influence to the system reliability. Identification and description of this influence should not be ignored in order to obtain relevant results of the reliability estimation of the systems which involve this kind of spares.

As it was abovementioned, the *shared redundancy* (SR) mentioned by Wysocki and others in [13] is in further text taken into account in the same meaning as a *quasi-redundant* (QR) component. Thus, quasi-redundant components are the parts of the system which follow their primary mission when the entire system is in functional state. However, when some parts of the system fail then this function could be replaced by another part which follows the same or a similar mission, thus by quasi-redundant part. The quasi-redundant components are not primary determined as active redundant subsystem because each one has its own mission which must be accomplished. Only in case of failure it could be used. In NCS appears the question of logical reconfiguration of the system when the data flow must be changed in order to replace the functionality of a subsystem by another one. For example, some new node will lose the network connection and system has to avoid the state when packets are sent to node which does not exist. Thus, the main features of the shared redundancy could be summarized as follows:

> *"Quasi-redundant component is not considered as primary redundant component such as the active or the passive redundant components."*

Generally in networked control systems, three kinds of quasi-redundant components (subsystems) could be considered:

- QR controllers.
- QR networks.
- QR sensors.

Hence, a necessary but not sufficient condition is that a control structure where SR could be considered has to be composed at least of two abovementioned subsystems (controllers, networks, actuators). The subsystems should have similar functionality or construction in order to be able to replace the mission of another component. In case of quasi-redundant components there are several limitations. In order to take profit of quasi-redundant networks, it is necessary to connect all nodes in all considered QR networks. Thus, in case of different networks the components should have implemented all necessary communication interfaces. In case of QR controllers the hardware performance has to allow implementing more than one control task.

Third mentioned components are sensors. Consideration of the sensors as QR components has important physical limitations. In order to be able to replace a sensor for measuring a physical value $X$ by another one for measuring $Y$ it is necessary to use "multi-functional" smart sensors. *We can suppose that some combination of the physical values can not be measured by using one sensor due to inability to implement required functionality in one hardware component.*

Other limitation is the distance between failed sensor and its QR sensor which could have a significant influence to the possibility of its replacing. Generally, implementation of the QR sensors within control system structure could be more difficult than the application of the SR approach on controllers or networks.

There are several naturally suitable control structures which could implement the shared redundancy approach without other modifications such as cascade control structure (Fig. 1). This structure is often used in industrial applications thanks to its important features which improve the quality of control. With using cascade control structure there are several constraints [13]. The main condition requires that controlled system must contain subsystem (secondary subsystem FS(s) – Fig. 1) that directly affect to primary system FP(s). Thus, cascade structure composes of two independent controllers which could be used in order to implement the shared redundant approach.

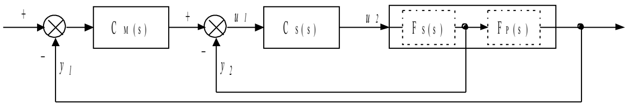


Fig. 1 Main structure of the cascade control

Usually for secondary subsystems there is a condition of faster dynamics than primary process. This condition must not be fulfilled [13] however, some modifications of conventional cascade structure (Fig. 1) and control laws must be provided.

### I. Quasi-redundant controllers

In previous text, several suitable control structures were briefly introduced. As was shown the controllers covered by these structures could be considered as quasi-redundant components by default. Thus, the hardware of both components could be shared in order to implement shared redundant approach.

Suppose the networked cascade control system shown in figure 2. The system is composed of five main components (Sensor $S_1$, $S_2$, controllers $C_1$, $C_2$ and actuator A) and two networks. The communication flow among components is determined by its cascade control structure. Thus, sensor $S_1$ sends a measured value to controller $C_1$ (*Master*), the controller $C_2$ (*Slave*) receives the values from the sensor $S_2$ as well as the controller $C_1$ in order to compute an actuating value for the actuator A.

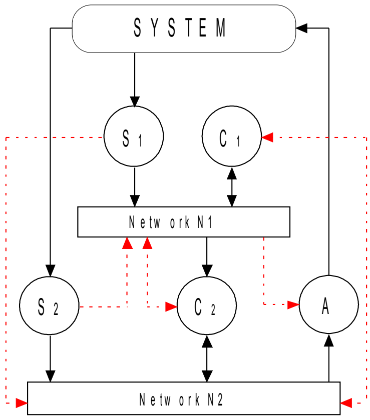


Fig. 2 NCCS with two networks and alternative network connections

Each part of the system (components and networks) presents independent subsystem. However, when quasi-redundant components are considered the system is not already composed of the independent components. Depending on the performance parameters of the used hardware equipment in the control loop, a specific influence on the system reliability should be taken into account. Thus some dependencies should not be ignored in the dependability analysis. In the NCCS shown in figure 2 we could consider controllers $C_1$ and $C_2$ as the quasi redundant subsystems (components). Both QR controllers have the primary mission which must be followed. Thus, controller $C_1$ controls outer control loop and controller $C_2$ stabilizes inner control loop. However in case of failure of one of them, we could consider the second one as some kind of spare.

As was abovementioned, the controllers follow their primary mission stabilization or performance optimization of the controlled system. Therefore, in regard to the similar hardware it allows sharing the computing capacity and executes different tasks. Thus, in order to implement SR approach both controllers *have to encapsulate both control tasks* – for the outer and the inner control loop (see the cascade control structure in figure 1).

In non-failure mode the primary task is executed in both controllers. However, in case of controller's failure (primary or secondary) non-failed controller starts execute both tasks and computes actuating value for primary as well as secondary subsystem. In this case we can suppose two scenarios.

The first one supposes that the controller is able to execute all the necessary tasks within the required sample periods (Fig. 3a). Thus, no delays or other undesirable consequences are expected. In this case the behavior of the quasi-redundant component is similar as in case of the active redundant components. Thus, in case of failure of one of the components, the second takes care about its mission until its failure.

Figure 3b shows a second case when time to execute both necessary tasks is greater than the required sampling period. Thus, the controller will cause the delays which have significant influence to the system stability [5] [7]. Therefore, this delay could be known that allows its partially compensating by using several methods [10]. Thus, we can suppose that system destabilization will not occur immediately after the first delay and we are able to compensate it for some time interval. Thus, quasi-redundant controller does not fail immediately but its reliability decreased.

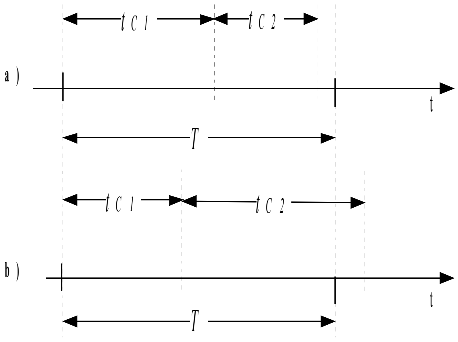


Fig. 3 Possible scenarios for quasi-redundant controllers

There are several situations when this scenario could be considered. In critical systems where failure of an important component could cause undesired damage or other dangerous consequences the shared redundancy approach could helps to allocate some time interval in order to take the system in a safe state. Thus, SR approach can be a significant technique how to save the system before damage.

### *II. Quasi-redundant networks*

The second part of the NCS which could be taken into account as an SR subsystem are networks. Suppose a system with two networks (Fig. 2) where all components could communicate (connect) on these networks ($N_1$ and $N_2$) if is it needed. In this case we can apply SR approach on this system.

Considered functionality of the quasi redundant networks is as follows. Both networks transmit required data - network $N_1$ transmit data from $S_1$ to $C_1$ and from $C_1$ to $C_2$ such as network $N_2$ from $S_2$ to $C_2$ and from $C_2$ to A. Thus both networks are active and allocated during the system mission. The same as in case of QR controllers, when one network has failed the second one can take its load after a system reconfiguration. Thus, all required data are sent through the second network. Hence, two similar scenarios as with controller task execution could be described. The amount of transmitted data on network with specified bit rate has logically influence on the probability of failure of the network (of course this depends on the network type and other parameters mentioned). This influence could be ignored when network performance parameters are sufficient. However, we can suppose that probability of network failure is increasing simultaneously with increasing network loading.

The characteristic between network loading and its bit rate depends on the network type and have to be measured in real network conditions in order to determine the type of dependency – linear or nonlinear.

Not only the network bit rate can be important however other network limitations such as maximal number of nodes connected to network, etc. All limits of the QR subsystems can create dependencies with direct influence on the system reliability. Primary, we could consider these dependencies as undesirable but in case of critical failures this SR approach gives some time to save the system.

When NCS with SR approach are analyzed this characteristic should be included in prepared model and further evaluated in order to determine its influence to the reliability of the whole NCS.

### *III. Different scenarios in shared redundancy*

When certain dependencies are ignored we could regard on the control system with QR components as control structure with active redundant components. However, there are several important scenarios when the reliability of the system could be decreased in order to prevent dangerous consequences or other undesirable events.

These scenarios could appear when some conditions could not be fulfilled (insufficient execution time or network bit rate) but the system need some time in order to take a safe state. Hence, it is necessary to identify and describe the influence of these dependencies which leads to more relevant results. Thus, prevent from too pessimistic or too optimistic results of the reliability analysis of the considered systems. The dependencies could be distinguished as follows:

- active redundant dependency,
- single step change of the nominal failure rate $\lambda_n \in \langle 0;1 \rangle$ - increased once by constant value – step load change,
- time depend change of the nominal failure rate $\lambda_n$ - functional dependency –the load of the subsystem is changed with time passed from speared subsystem failure,
    - linear,
    - nonlinear.

We suppose the presumption that destabilization of the system does not occur immediately after the first delay on the network caused by insufficient controller's hardware or network's parameters. Thus, quasi-redundant controller does not fail immediately but in this case its failure rate increases which correspond consequently to a decreased reliability.

Thus, in case of the active redundant dependency we suppose that quasi-redundant subsystem has sufficient capacities in order to follow its primary mission as well as the mission of the failed subsystem (or subsystems).

Single step change of the nominal failure rate of the subsystem is considered in case of subsystems where the failure rate of the quasi-redundant subsystem is changed (increased) once by constant value (Fig. 4) during its life time. Thus, the new increased failure rate $\lambda$' remains constant during further life time of the subsystem. For example, let's suppose a NCS with two Ethernet networks where one of them has failed and consequently the system is reconfigured and all nodes (components) start to communicate through the non-failed network which has sufficient bit rate capacity in order to transmit all required data. However, the amount of data has been increased which consequently increases the probability of packets' collisions. Thus, probability of the failure (failure rate) has been increased up to new value $\lambda$'.

A third case considers the change of the nominal failure rate $\lambda_n$ which depends on the time passed from the moment of the failure until current time of the working of the quasi-redundant subsystem which encapsulates the executing necessary tasks (own tasks as well as tasks of the failed subsystem). Thus, a functional dependency has to be considered. This dependency of the change of the failure rate $\lambda_n$ could be described by linear or nonlinear dependency / function. We could take previous example of the system with two networks. However, in this case the bit rate of the second (non-failed) network is not sufficient. Consequently delays in data transmission as well as other consequential undesirable problems such as system destabilization might be caused. We can suppose that the non-failed network will fail in some time. Thus, the nominal failure rate $\lambda_n$ of the second network is now time dependent and is linearly or nonlinearly increased until system failure. Mentioned examples with related equations are further discussed in more details.

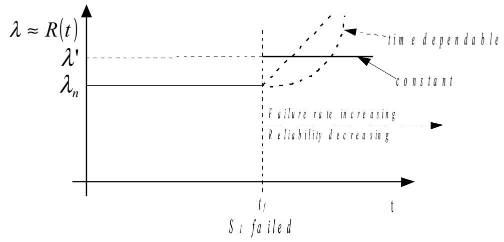


Fig. 4 Possible failure rate curves for subsystem S2 during its mission

Let's suppose that the reliability of the system $R(t)$, probability of the failure during time interval <0; $t$>, is characterized by a nominal failure rate $\lambda_n \in \langle 0;1 \rangle$ . Let's suppose a system with two subsystems $S_1$ and $S_2$ (such as networks in abovementioned examples) whereas the subsystem $S_1$ will fail as first and then quasi-redundant subsystem $S_2$ will follow both missions ($S_1$ and $S_2$). In figure 4 are shown two above mentioned scenarios when the nominal failure rate $\lambda_n$ of the subsystem is increased by a constant value or by value which could be described as linear or nonlinear function (functional dependencies).

At first increasing the failure rate $\lambda_n$ one time by constant value (see Fig. 4) will be dealt. It corresponds with the reliability reduction of the quasi-redundant subsystem $S_2$ by increasing the failure rate, during its mission, from its nominal value $\lambda_n$ up to new $\lambda$'. Consequently, the system will follow its primary mission thanks to QR subsystem $S_2$ but its failure rate is already increased and consequently the probability of failure of $S_2$ is higher. The difference between nominal $\lambda_n$ and increased $\lambda$' failure rate will be called *decrease factor* $d_R$. Thus, mentioned constant value is characterized by decrease factor $d_R$ of QR subsystem and new changed failure rate $\lambda$' at fail time $t_f$ is given by followed simple formula

$$\lambda' = \lambda_n + d_R \tag{1}$$

Failure rate increases only one time by the specified value and QR subsystem $S_2$ with new constant failure rate $\lambda$' will follow both mission of its own mission and mission of the failed subsystem $S_1$.

The second case shown in figure 3 considers reliability reduction where the failure rate $\lambda_n$ is increased during the working of the subsystem $S_2$ by a specified *decrease factor*. This change of the nominal failure rate depends on time whereas with time extending the failure rate of the $S_2$ is got near to 1 (system failed). Thus, a *decrease function* $f_{dr}(t)$ is represented by linear or nonlinear characteristic and depends on real subsystem which is considered as quasi-redundant. Thus, increased failure rate $\lambda$' of the subsystem $S_2$ depends on time $t$ and is given by following formula:

$$\lambda'(t) = \lambda_n + f_{dR}(t) \, . \tag{2}$$

As it was mentioned, the decrease function $f_{dr}(t)$ can be represented by a simple linear function, for example,

$$\lambda'(t) = \lambda_n + d_R 10^{-3}(t + 1 - t_f) \tag{3}$$

where $t$+1 allows change the nominal failure rate $\lambda_n$ at the moment of the failure at time $t_f$.

On the other side a nonlinear exponential function can be considered as follows:

$$\lambda'(t) = \lambda_n + e^{d_R(t-t_f)} \tag{4}$$

where $\lambda$' is the value of the increased failure rate, $\lambda_n$ is the nominal failure rate of the component, $t_f$ is the time of the failure of the component, $d_R$ is the decrease factor which has a direct influence on the increased failure rate.

### *IV. Application to a mini-drone helicopter*

The NCC structure is applied for the control of a four rotors mini-helicopter (Drone, Fig. 5). The proposed control structure for this real model is as follows. The NCC architecture is composed of one primary controller (Master) and one secondary controller (Slave), thirteen sensors, four actuators and two communication networks.

The Master is designed for attitude stabilization (control) through Slave controller for angular velocity control for each propeller. The aim of the control is to stabilize coordinates of the helicopter [11].

The controllers are used as quasi-redundant components within presented networked cascade control system (further only NCCS). They use the same control algorithm (propeller's angular velocity control) but with different input data (set point, system output, etc.)

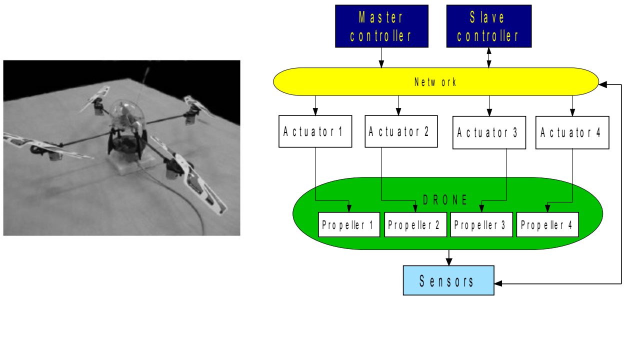


Fig. 5 Cascade control structure of mini-helicopter with one network

Hence, in case of failure one of them could retransmit all required data to another one, whereas pre-programmed control algorithm should compute the actuating value. Thus, failed controller is replaced by second one which start to compute actuating value.

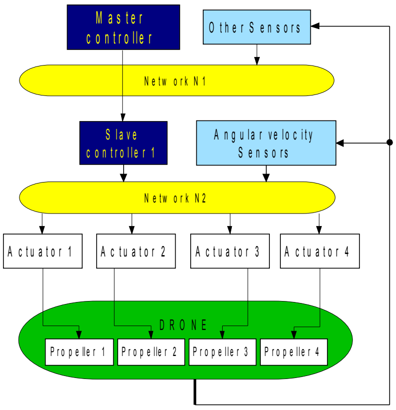


Fig. 6 Cascade control structure of mini-helicopter with two networks

Other quasi-redundant parts of this control structure are networks (Fig. 6). The same as in case of controllers, one of the networks can compensate another one after system reconfiguration. Usually, two networks are primary designed due to reduction amount of transmitted data. However, in case of network failure all data could be retransmitted through second one.

Described approach for subsystem's failure compensation by using the shared redundancy requires logical reconfiguration of the NCCS. Thus, in case of failure the hardware configuration is non-touched but communication ways must be changed in order to transmit the data to non-failed component or through non-failed network.

## IV. Simulation and Results

All presented networked control architectures (Fig. 5, 6) were modelled by using Petri nets. This tool was chosen thanks to its ability to model different types of complex systems and dependencies within them. To provide the reliability analysis the Monte Carlo simulation (further only MCS) method was used. The multiple simulations of the modelled architecture [ 12] are provided to obtain the reliability behavior the basic two quasi-redundant components (for example two controllers in CCS structure).

Model of the system covers the simulation of the random events of the basic components of the system such as sensors, controllers and actuators as well as the network's random failures. Software used for model preparation is CPN Tools which allow multiple simulation of the model in order to obtain statistically representative sample of the necessary data to determine the reliability behaviour of the studied model.

As was mentioned, the simulation of the simple two quasi-redundant components with all considered changes of the failure rate (single, linear, nonlinear) was provided. Thus, new failure rate $\lambda$' of the non-failed component is computed by using equation (1), (3) and (4).

TABLE IV.
MTTFF OF SIMULATED CONTROL STRUCTURES WITH DIFFERENT DECREASE FACTORS

| Decrease factor – $d_R$ | MTTFF - Drone (Fig. 5) | MTTFF -Drone (Fig. 6) |
|---|---|---|
| 0 | 55 (+11%) | 58 (+22%) |
| $2.10^{-3}$ | 54 (+9%) | 56 (+17%) |
| $10^{-2}$ | 53 (+7%) | 54 (+13%) |
| $59.10^{-2}$ | 50.5 (+2%) | 49 (+3%) |
| 0.999 | 49.6 | 47.6 |

This change could be called as single change because the component's failure rate is changed only once during QR component's life time. Both components has equal nominal failure rate $\lambda_n = 0.001$.

Few examples of the influence of the single step change of the failure rate by the specified decrease factor $d_R$ to the reliability behaviour are shown in figure 7. We can see there are five curves. Two non-dashed curves show studied system as system with two active redundant components (thus, $d_R$ is equal to zero – first curve from the top) and as system without redundant components (thus, system composes of two independent components without redundant relation – first curve from the bottom). These two curves determine borders where reliability of the studied system can be changed depending on value of the decrease factor $d_R$.

As we can see from figure 7, single increasing of the nominal failure rate $\lambda_n$ of the non-failed components by the same value as was nominal failure rate $\lambda_n$ up to $\lambda' = 0.002$ ($d_R = 0.001$) cause significant reduction of the reliability.

In tables I, are shown several values of life time (parameter MTTFF) for this studied system. Each table (Table I, II, III) shows the life time of the studied components as active redundant subsystems ($d_R = 0$) and as independent subsystems ($d_R = 0.999$). From value of the decrease factor $d_R = 0.01$ the life time of the system significantly improves (18% and more). The results of the linear and nonlinear failure rate increasing are shown in tables II and III. In all tables are noted the percentual value of the increased life time corresponding to the decrease factor.

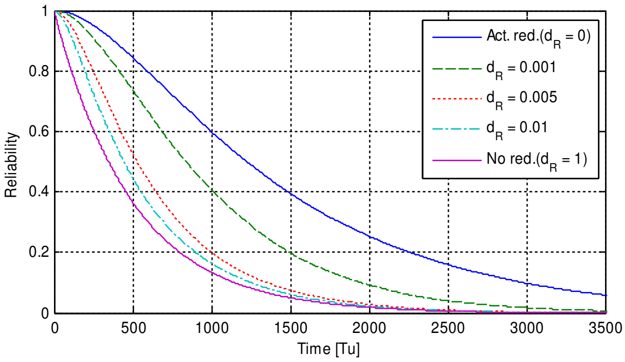


Fig. 7 Influence of the increased failure rate of the component by constant decrease factor $d_R$ to reliability of the system composed of two quasi-redundant components

Table IV shows the MTTFF parameters of both complex mini-helicopter structures. In the first drone structure (Fig. 5) two quasi-redundant controllers are considered. In the second structure (Fig. 6) two groups of quasi-redundant subsystems are considered and simulated – the controllers and the networks.

In all simulated systems was observed the influence of the single step of the failure rate by a value specified by the decrease factor $d_R$. The same as in tables I – III, there are shown the life time of system corresponding to different decrease factors $2.10^{-3}$, $10^{-2}$, $59.10^{-3}$. We can see that increasing the component's nominal failure rate $\lambda_n$ by decrease factor equal to $59.10^{-3}$, which represents approximately 59 times higher failure rate, has a significant influence to decreasing the life time of the system. The results are a little bit better than in the case of the system

TABLE I.
MTTFF OF THE TWO QUASI-REDUNDANT WITH SINGLE STEP CHANGE OF THE FAILURE RATE

| $\lambda_n = 10^{-3}$ | Act. red. $d_R = 0$<br>($\lambda' = 10^{-3}$) | $d_R = 0.001$<br>($\lambda' = 0.002$) | $d_R = 0.005$<br>($\lambda' = 0.006$) | $d_R = 0.01$<br>($\lambda' = 0.011$) | $d_R = 0.1$<br>($\lambda' = 0.101$) | No red. $d_R = 0.999$<br>($\lambda' = 1$) |
|---|---|---|---|---|---|---|
| MTTFF [*Tu*] | 1503 (+300%) | 1002 (+200%) | 667 (+34%) | 589 (+18%) | 509 (+2%) | 499 |

TABLE II.
MTTFF OF THE TWO QUASI-REDUNDANT WITH LINEAR INCREASING OF THE FAILURE RATE

| $\lambda_n = 10^{-3}$ | Active red.<br>$d_R = 0$ | $d_R = 10^{-3}$ | $d_R = 10^{-2}$ | $d_R = 10^{-1}$ | No redundancy |
|---|---|---|---|---|---|
| MTTFF [*Tu*] | 1503 (+300%) | 1153 (+231%) | 812 (+63%) | 611 (+22%) | 499 |

TABLE III.
MTTFF OF THE TWO QUASI-REDUNDANT WITH EXPONENTIAL INCREASING OF THE FAILURE RATE

| $\lambda_n = 10^{-3}$ | Active red.<br>$d_R = 0$ | $d_R = 10^{-3}$ | $d_R = 10^{-2}$ | $d_R = 10^{-1}$ | No redundancy |
|---|---|---|---|---|---|
| MTTFF [Tu] | 1503 (+300%) | 902 (+80%) | 676 (+35%) | 537 (+8%) | 499 |

without redundant components ( $d_R = 0.999$), but we could say that they are almost the same.

The drone's structure composes of twenty (twenty-one – structure with two networks) components – thirteen sensors (3 gyrometers, 3 magnetometers, 3 accelerometers, 4 rotors' angular velocity sensors), two controllers, four actuators and one (two) networks. Due to high ratio of the independent components and shared redundant components within drone's structure (18 independent and 2 quasi-redundant – Fig. 5) there is a difference between life times for minimal and maximal $d_R$ is significantly smaller (about 11% and 22%) than in case of basic two components subsystem (Table I, II, III).

The Mean Time Before First system's Failure is significantly longer in case of basic two component subsystem than in drone's cases. As it was mentioned above this is caused by the difference in complexity between basic and drone's NCC architecture. In case of comparison between two drones structures (Fig. 5, 6) the results are better for architecture with two networks which is composed of two quasi-redundant subsystems – controllers (Master, Slave) and networks when the decrease factor is smaller than $59.10^{-3}$ . The increasing of the nominal failure rate by the decrease factor greater than $59.10^{-3}$ significantly decreases the life time of the drone. On the other side, even if the controller loading will change its failure rate approximatelly ten times ( $d_R = 10^{-2}$ ) the system's life time is about 7% longer than in case of the system without shared redundant approach implementation.

## V. Conclusion

The paper shows the influence of additional reliability decreasing of the quasi-redundant component to entire reliability of the studied system. Description of this dependency is getting closer to show the behavior of the system reliability when shared redundancy approach is implemented . The results shown in tables I – III could be very helpful in order to approximate the life time of the quasi-redundant subsystem under different conditions of the failure rate increasing. Presented cascade control architecture suitable for shared redundancy approach implementation could be applied to similar systems. For example, Steer-by-Wire control [9] of two front wheels in a car, etc. In addition the paper has shown the conventional cascade control structure within conditions of networked control systems as naturally suitable to profit from quasi-redundant subsystems as networks, controllers and potentially sensors if physical process allows it. Despite of some constraints for using this type of control, cascade architecture is widely used in industrial control applications. Hence, only the reconfiguration algorithm should be implemented to take profit from quasi-redundant subsystems.

The main advantages of the quasi-redundant components could be summarized as follows:

- The system is composed only of necessary components (parts) for following the primary mission of the system whereas higher system reliability is ensured without using any additional active redundant components.
- Following the first point we could suppose less number of components used for saving the control mission. Thus, economic aspect could be significant.
- Prevention of the system's critical failure when QR subsystem has no sufficient hardware capacities.

## References

[1] M. Bebbington, C-D. Lai, R. Zitikis, "Reliability of Modules with Load Sharing Components", *Journal of Aplied Mathematics and Decision Sciences* , 2007.

[2] C. Brosilow, J. Babu, *Techniques of Model-Based Control*, Prentice Hall, 2002, ch. 10.

[3] P. Castillo, A. Dzul, R. Lozano, "Real-Time Stabilisation and Tracking of a Four Rotor Mini-Rotorcraft", *IEEE Transaction on control systems technology*, Vol. 12, No. 4, 2004, pp. 510 – 516.

[4] F. Guenab, D. Theilliol, P. Weber, Y.,M. Zhang, D., "Sauter, Fault-tolerant control system design: A reconfiguration strategy based on reliability analysis under dynamic behaviour constraints", *6th IFAC Symposium on Fault Detection*, 2006, pp. : 1387-1392.

[5] J. Galdun, R. Ghostine, J. M. Thiriet, J. Liguš, J. Sarnovský, "Definition and modelling of the communication architecture for the control of a helicopter-drone", *8th IFAC Symposium on Cost Oriented Automation*, 2007.

[6] J. Galdun, J. Liguš, J-M. Thiriet, J. Sarnovský, "Reliability increasing through networked cascade control structure – consideration of quasi-redundant subsystems", *World IFAC Congress*, Seoul, South Korea, 2008.

[7] J. Ligušová, J.M. Thiriet, J. Liguš, P. Barger, "Effect of Element's Initialization in Synchronous Network Control System to Control Quality", *RAMS/IEEE conference Annual Reliability and Maintainability Symposium*, 2004.

[8] J. C. Laprie, , H. Kopetz, A. Avižienis, (1992). Dependability: Basic Concepts and Terminology, Chapter 1, Springer-Verlag / Wien, ISBN: 3-211-82296-8.

[9] G. Leen, D. Heffernan, "Expanding Automotive Electronic Systems", *Computer IEEE*, Vol. 35, 2002, pp.: 88-93.

[10] S.,I. Nicolescu,. *Stabilité systèmes à retard – Aspects qualitatifs sur la stabilité et la stabilisation* , Diderot multimedia, 1997.

[11] P. Pozsgai, W. Neher, B. Bertsche, "Models to Consider Load-Sharing in reliability Calculation and Simulation of Systems Consisting of Mechanical Components", *IEEE – Proceedings annual reliability and maintainability symposium* , 2003, pp.: 493 – 499.

[12] J. T. Spooner, K., M. Passino, "Fault-Tolerant Control for Automated Highway Systems", *IEEE Transactions on vehicular technology*, vol. 46, no. 3, 1997, pp. 770-785.

[13] J. Wysocki, R. Debouk, K. Nouri, "Shared redundancy as a means of producing reliable mission critical systems", *2004 Annual Symposium – RAMS - Reliability and Maintainability*, 2004, pp.: 376-381.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Runtime resource assurance and adaptation with Qinna framework: a case study

Laure Gonnord*, Jean-Philippe Babau
CITI / INSA-Lyon
F-69621 Villeurbanne Cedex – France
Email: {Laure.Gonnord, Jean-Philippe.Babau}@insa-lyon.fr

***Abstract*—Even if hardware improvements have increased the performance of embedded systems in the last years, resource problems are still acute. The persisting problem is the constantly growing complexity of systems. New devices for service such as PDAs or smartphones increase the need for flexible and adaptive open software. Component-based software engineering tries to address these problems and one key point for development is the Quality of Service (QoS) coming from resource constraints. In this paper, we recall the concepts behind Qinna, a component-based QoS Architecture, which was designed to manage QoS issues, and we illustrate the developpement of a image viewer application whithin this framework. We focus on the general developpement methodology of resource-aware applications with Qinna framework, from the specification of resource constraints to the use of generic Qinna's algorithms for negociating QoS contracts at runtime.**

## I. Introduction

THE STUDY takes place in the context of embedded handled systems (personal digital assistants, mobile phones) whose main characteristic is the use of limited resources (CPU, memory).

In order to develop multimedia software on such systems where the quality of the resource (network, battery) can vary during use, the developer needs tools to:

- easily add/remove functionality (services) during compilation or at runtime;
- adapt component functionality to resources, namely propose "degraded" modes where resources are low;
- evaluate the software's performances: quality of provided services, consumption rate *for some scenarios*.

In this context, component-based software engineering appears as a promising solution for the development of such kinds of systems. Indeed it offers an easier way to build complex systems from base components ([1]), and thus we are able to design resource components like others. The main advantages are the re-usability of code and also the flexibility of such systems.

The Qinna framework ([2], [3]) was designed to handle the specification and management of resource constraints problems during the component-based system development. Variability is encoded into discrete implementation levels and links between them. We can also encode quantity of resource constraints. Qinna provides algorithms to ensure resource constraints and dynamically adapt the implementation levels according to resource availability *at runtime*.

In this paper, we present a case study using Qinna as proof of concept. In Section II we present the main characteristics of the case study which is an image remote viewer. In Section III we recall Qinna's main concepts, as introduced in [2] and formalize them in a more generic way. We give an overview of Qinna's C++ implementation (Section IV), and then provide the general implementation steps to develop a resource-aware application with Qinna (Section V). We illustrate in the particular case of the remote viewer application in Section VI.

## II. Specification of the remote viewer

Our case study is a remote viewer application whose high level specification follows:

- The system is composed of a mobile phone and a remote server. The application allows the downloading and the visualization of remote images via a wireless link.
- The remote directory is reached via a ftp connection. After connection, two buttons "Next" and "Previous" are used to display images one by one. Locally, some images are stored in a buffer. To provide a better quality of service, some images are downloaded in advance, while the oldest ones are removed from the photo memory.
- The application must manage different qualities of services for the resources: shortage of bandwidth and memory, or disconnections of the ftp server. When needed it can download images in lower quality (in size or image compression rate).
- Different storage policies are possible, and there are many parameters which can be modified; like the size of the buffer, or the number of images that are downloaded each time. We want to evaluate which policy is the best *according to a given scenario*.

We aim to use Qinna for two main objectives: maintenance of the application with respect to the different qualities of service, and also the evaluation of the influence of the parameters on the non-functional behavior (timing performance and resource usage) of the application.

## III. Description of the Qinna framework

### A. Qinna's main concepts

The framework designed in [2] and [3] has the following characteristics:

[*] This work has been partially supported by the REVE project of the French National Agency for Research (ANR)

- Both the application pieces of code and the resource are components. The resource services are enclosed in components like `Memory`, `CPU`, `Thread`.
- The variation of quality of the provided services are encoded by the notion of *implementation level*. The code used to provide the service is thus different according to the current implementation level.
- The link between the implementation levels is made through an explicit relation between the implementation level of the provided service and the implementation levels of the services it requires. For instance, the developer can express that a video component provides an image with highest quality when it has enough memory and sufficient bandwidth.
- All the calls to a "variable function" are made through an existing contract that is negotiated. This negotiation is made automatically through the Qinna components. A *contract* for a service at some objective implementation level is made only if all its requirements can be reserved at the corresponding implementation levels and also satisfy some constraints called Quality of resource constraints (QoR). If it not the case, the negotiation fails.

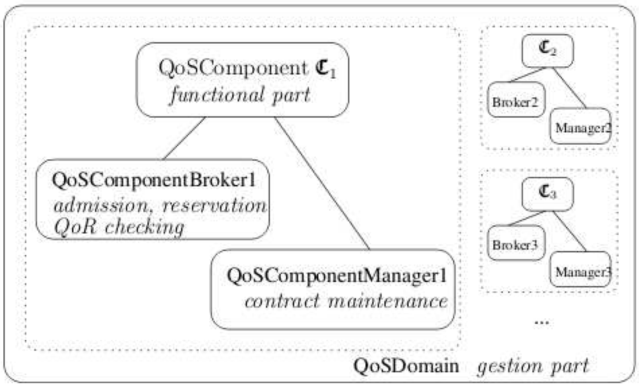


Fig. 1. Architecture example

These characteristics are implemented through new components which are illustrated in Figure 1: to each application component (or group of components) which provide one or more variable service Qinna associates a *QoSComponent* $\mathfrak{C}_i$. The variability of a variable service is made through the use of a corresponding `implementation level` variable. Then, two new components are introduced by Qinna to manage the resource issues of the instances of this *QoSComponent*:

- a *QoSComponentBroker* which goal is to realize the admission of a component. The Broker decides whether or not a new instance can be created, and if a service call can be performed w.r.t. qthe uantity of resource constraints (QoR).
- a *QoSComponentManager* which manages the adaptation for the services provided by the component. It contains a mapping table which encode the relationship between the implementation levels of each of these services and their requirements.

At last, Qinna provides a single component named *QoSDomain* for the whole architecture. It manages all the service requests inside and outside the application. The client of a service asks the Domain for reservation of some implementation level and is eventually returned a contract if all constraints are satisfied. Then, after each service request, the Domain makes an acknowledgment only of the corresponding contract is still valid.

### B. Quantity of Resource constraints in Qinna

A Quantity of resource constraint (QRC) is a quantitative constraint on a component $\mathfrak{C}$ and the service $(s_i)$ it proposes. QRCs are for instance formula on the total instance of a given component type, of the total amount of resource (memory, CPU) allocated to a given component. They are two types of constraints, depending on their purpose:

- Component type constraints (CTC) express properties of components of the same type and their provided services.
- Component instance constraints (CIC) express properties of a particular instance of a component.

The management of these constraints is automatically done at runtime, if the developer implements them in the following way:

- In the `QoSComponent`, for each service, implement the two functions: `testCIC` and `updateCIC`. The former decides whether or not the call to the service can be performed, and the later updates variables after the function call. In addition, there must be an initialization of the CICs formulas at the creation of each instance.
- Similarly, in the `QoSComponentBroker`, for each provided service, implement the two functions `testCTC` and `updateCTC`.

Then, Qinna maintains resource constraints at runtime through the following procedure:

- When the Broker for $\mathfrak{C}$ is created, the parameters used in `testCTC` are set.
- The creation of an instance of $\mathfrak{C}$ is made by the Broker iff $CTC_{compo}(\mathfrak{C})$ is true. During the creation, the CIC parameters are set.
- The $CIC(s_i)$ and $CTC(s_i)$ decision procedures are invoked at each function call. A negative answer to one of these decision procedures will cause the failure of the current *contract*. We will detail the notion of contract in Section III-D.

### C. QoS Linking constraints

Unlike quality of resource constraints, linking constraints express the relationship between components, in terms of quality of service. For instance, the following property is a linking constraint: " to provide the `getImages` at a "good" level of quality, the `ImageBuffer` component requires a "big" amount of memory and a "fast" network". This relationship between the different QoS of client and server services are called QoS Linking Service Constraints (QLSC).

**Implementation Level**To all provided services that can vary according to the desired QoS we associate an *implementation* *level*. This implementation level (IL) encodes which part of implementation to choose when supplying the service. These implementation levels are totally ordered for a given service. As these implementation levels are finitely many, we can restrict ourselves to the case of positive integers and suppose that implementation level $0$ is the "best" level, $1$ gives a lesser quality of service, and so on.

We assume that required services for a given service doesn't change according to the implementation level, that is, the call graph of a given service is always the same. However, the arguments of the required services calls may change.

**Linking constraints expression** Let us consider a component $\mathfrak{C}$ which provides a service $s$ that requires $r_1$ and $r_2$ services. Qinna permits to link the different implementation levels between callers and callees. The relationship between the different implementation levels can be viewed as a function which associates to each implementation level of $s$ an implementation level for $r_1$ and for $r_2$:

$$QLSC_s : \left| \begin{array}{ccc} \mathbb{N} & \longrightarrow & \mathbb{N}^2 \\ IL & \longmapsto & (IL_1, IL_2) \end{array} \right.$$

Thus, as soon as an implementation level is set for the $s$ service, the implementation level of all required services (and all the implementation levels in the call tree) are set. This has a consequence not only on the code of all the involved services but on the arguments of the service calls as well.

Therefore, if a user asks for the service $s$ at some implementation level, the request may fail due to some behavioral constraint. That's why every request for a service must be negotiated and the notion of contract will be accurate to implement a set of a satisfactory implementation levels for (a set of) future calls.

**Implementation of linking constraints in Qinna** The links between the provided QoS and the QoS of the required services are made through a table whose lines encode the tuples of linked implementation levels: $(IL_s, IL_{r_1}, IL_{r_2})$. This "mapping" table is encoded in the QoSManager. The natural order of the lines of the table is used to determine which tuple to consider if the current negotiation fails.

Now we have all the elements to define the notion of contract.

### D. Qinna's contracts

Qinna provides the notion of *contract* to ensure both behavioral constraints and linking constraints.

When a service call is made at some implementation level, all the subservices implementation level are fixed implicitly through the linking constraints. As all the implementation levels for a same service are ordered, the objective is to find the best implementation level that is feasible (w.r.t. the behavioral constraints of all the components and service involved in the call tree).

**Contract Negotiation** All service calls in Qinna are made after negotiation. The user (at toplevel) of the service asks for the service at some interval of "satisfactory" implementation levels. Qinna then is able to find the best implementation level in this interval that respects all the behavioral constraints (the behavioral constraints of all the services involved in the call tree). If there is no intersection between feasible and satisfactory implementation levels, no contract is built. In the other case, a contract is made for the specific service. A contract is thus a tuple $(id, s_i, IL, [IL_{min}, IL_{max}], imp)$ denoting respectively its identifiant number, the referred service, the current implementation level, the interval of satisfactory implementation levels, and the *importance* of the contract. This last variable is used to sort the list of all current contracts and is used for degradation (see next paragraph).

After contract initialization, all the service calls must respect the terms of the contract. In the other case, there will be some renegotiation.

**Contract Maintenance and Degradation** After each service call the decision procedure for behavioral constraints are updated. After that, a contract may not be valid anymore. As all service calls are made through the Brokers by the Domain, the Domain is automatically notified of a contract failure. In this case, the Domain tries to degrade the contract of least importance (which may be not the same as the current one). This degradation has consequences on the resource and thus can permit other service calls inside the first contract.

Basically, degrading a contract consists in setting a lesser implementation level among the satisfactory ones, but which is still feasible. If it is not possible, the contract is stopped.

**Use of services** Each call to a service at toplevel as consequences on the contract which has been negociated for him. We suppose that a contract is made before the first invocation of the desired service. The verification could automatically be done with Qinna, but is not not yet implemented. All the notifications of failures are logged for the developer.

## IV. Qinna's components implementation in C++

We implemented in C++ the Qinna components and algorithms. These components are provided through classes which we detail in this section.

### A. Qinna's components for the management of services

**QoSComponent** The QoSComponent class provides generic constructors and destructors, and contains a private structure to save the current implementation levels of the component provided service. All QoS components will inherit from this class.

**QoSBroker**The QoSBroker class contains a private structure to save the references to all the corresponding components it is responsible for. It provides the two functions `Free(QoSComponent* refQc)` and `Reserve(...)`. As `testCIC` and `updateCIC` functions signature depends of each component/service, these functions will be provided in each instance of QoSBroker.

**QoSManager** The QoSManager class contains all information for the service provided by its associated component. It provide the following public functions:

- `bool SetServiceInfos(int idserv, QoSComponent *compo, int nbreq, int nbmap)` initializes the manager for the *idserv* service, provided by **compo*, with *nbreq* required services and *nbmap* different implementation levels. Return `true` if successful, `false` otherwise.
- `bool AddLevQoSReq(int idserv, int lv, int irq, int lrq)` adds the tuple $(lv, irq, lrq)$ (the $lv$ implementation level for $idserv$ is linked to the $lrq$ implementation level for $irq$ service) in the mapping table for $idserv$.
- `int Reserve(int idserv, int lv)` is used for the reservation of the $idserv$ service at level $il$. It returns the local number of (sub) contract of the Manager or $0$ if the reservation has failed (due to resource constraints).

**QoSDomain** The QoSDomain class provides functions for managing contracts at toplevel:

- `bool AddService(int service, int nbRq, int nbMp, QoSManager *qm)` adds the service $service$ with $nbRq$ required services and $nbMp$ implementation levels, with associated manager $*qm$.
- `int Reserve(QoSComponent *compo,int ns , int lv, int imp)` is used for reservation of the service $ns$ provided by the component $*compo$ at level $lv$ and importance $imp$. it returns the number of contract (in domain) if successful, $0$ otherwise.
- `bool Free(int id)` frees the contract number $id$ (of domain).

**ManagerContract** This class provides a generic structure for a subcontract which encodes a tuple of the form $< id, lv, *rq, v >$ where $id$ is the contract number, $lv$ the current level, $rq$ is the component that provides the service and $v$ is a C++-vector that encode the levels of the required services. This class provides access functions to these variables and a function to change the implementation level.

**DomainContract** This class provides a structure for contracts at toplevel. A Domain contract is a tuple of the form $< di, i, lv, *rq >$ where $di$ is the global identifier of the contract, $*rq$ is the manager associated to the component that provides the service, $i$ is the local number of subcontract for the manager, and $lv$ is the current level of the service.

### *B. Basic resource components*

In the call graph of one service, leaves are physical resources (Memory, CPU, Network). As all resources must be encapsulated inside components, we need to encapsulate the base functions into QoSComponents. For instance, the `Memory` component must be encoded as a wrapper around the `malloc` function, and the associated broker basically implements the CIC functions which decide if the global amount of allocated memory is reached or not.

Sometimes, the basic functions are encapsulated in higher level components. For instance, a high level library might provide a `DisplayImage` function which makes an explicit call to `malloc`, but this call is hidden by the use of the library. In this particular case, the management of basic resource functions can be done in two different but equivalent ways:

- the creation of a “phantom” Memory component which provides the two services `amalloc` (for abstract malloc) and `afree`. Each time the developer makes a call to an “implicit” resource function (*i.e.* when the called function needs a significant amount of memory, like `DisplayImage`), he has to call `Memory.amallloc`. The Qinna’s C++ implementation provides some basic components like Memory, Network and CPU and their associated brokers.
- the creation of QoSComponent around the library function `DisplayImage` which is responsible (through its broker) for the global amount of “quantity of resource” used for the `DisplayImage` function.

Both solutions need a precise knowledge of the libraries functions w.r.t the resource consumption. We assume that the developer has this knowledge since he designs a resource-aware application. In our case study we used the first solution.

## V. Methodology to use Qinna

We suppose that in the application all resources, including hardware resources (Memory, CPU) or software ones (viewer, buffer), are encoded by components. Here are the main steps for integrating Qinna into an existing application designed in C++:

1) **Identify the variable services** which are functions whose call may fail due to some resource reasons. They are of two types:
   - simple functions like `Memory.malloc` whose code does not vary. They have a unique implementation level.
   - “adaptive” functions whose code can vary according to implementation levels.

   The first step is thus to identify the services whose quality vary and associate to each of this services a *unique* key, and if the code vary, clearly identify the variant code through a code of the form:

```
switch(implLevel)
   {
    case 0 :
       ...
   }
```

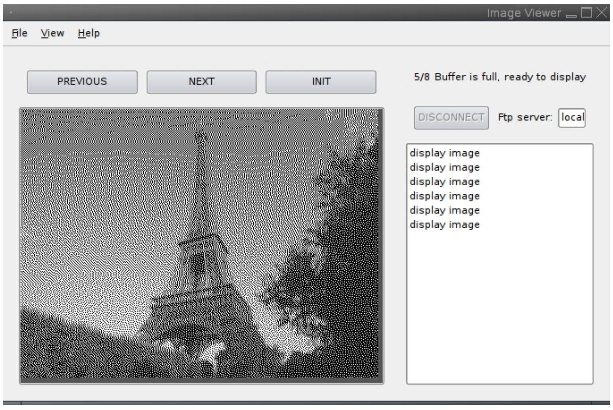
   where implLevel is the associated (variable) attribute of the host component for this service. We must identify which variable services are required for each provided service, and the relationship between the different implementation levels.

2) **Create Qinna components**. First, cut the source code into QoSComponents that can provide one or more QoSservices. As the QoS negotiation will only be made between QoSComponents of different types, this split will have many consequences on the QoS management. For each QoSComponentC (which inherits from the `QoSComponent` class), the designer must encode two classes: `QoSBrokerC` and `QoSManagerC` which respectively inherit from the `QoSBroker` and `QoSManager` generic classes. For the whole application, the designer will directly use the `QoSDomain` generic class.
3) **Implement Quality of Resource constraints**. These constraints are set in two different ways:
   - The type constraints (CTC) for component $C$ implementation is composed of additional functions in $QoSBrokerC$ : `initCTC` which is executed at the creation of the Broker, and which sets the decision procedures parameters ; a `testCTC` function to determine whether a new instance can be created or not ; an `updateCTC` to save modifications of the resources after the creation. For each provided QoS service $s_i$, we add to new functions: `testCTC(idsi)` which is executed before the call of a service and tells if the service can be done, and `updateCTC(idsi)` to be executed after the call.
   - The instance constraints (CIC) for $C$ are also composed of three functions to encode in the $QoSComponentC$: `setCIC` to set the resources constants, `testCTC(idsi)` which is used to decide if a service of identifiant `ids` can be called, and `updateCTC(idsi)` to update the resource constraints after a call to the $s_i$ function.
4) **Implement the linking constraints**. The links between required services and provided service via implementation levels are set by the invocation of the `SetService` and `AddLevQoSReq` functions of the managers. These functions will be invoked at toplevel.
5) **Modify the main file to initialize Qinna components at toplevel**. Here are the main steps:
   - For each base resource (CPU, Memory, ...)
     a) Invoke the constructor for the associated Broker. The constructor's arguments must contain the initialization of internal variables for type constraints (the total amount of memory for example).
     b) Create the associated Manager with the Broker as argument.
     c) Register the QoS services inside the Manager with call to the `SetServiceInfos` function.
     d) Create QoSComponents instances via the use of the `Broker.reserve(...)` function. The arguments can be a certain amount of resource used by the component.
   - For all the other QoSComponents, the required components first:
     a) Create the associated Broker and Manager.
     b) Set the services information.
     c) If a service requires another service of another component, use the function `Manager.AddReq` to link the required manager. Then use `Manager.AddLevQoSReq` to set the linking constraints.
     d) Create QoSComponent instances by invoking the corresponding reservation function (`Broker.Reserve`).
   - Create the QoSDomain and add the services that are used at toplevel (`Domain.AddService`)
   - Reserve services via the QoSDomain and save the contracts' numbers.

## VI. Viewer Implementation using Qinna

This case study is a proof of concept for using Qinna. For this specific application, we want to use Qinna for two objectives:

- the maintenance of the application with respect to the different qualities of service,
- the evaluation of the influence of the parameters on the non-functional behavior (timing performance and resource usage)



5/8 Buffer is full, ready to display

Ftp server: local

display image
display image
display image
display image
display image
display image

Fig. 2. Screenshot of the viewer application

### A. The functional part

The functional part of the viewer is developed with Qt[1] (a C++ library which provides graphical components and implementations of the ftp protocol). Figure 3 describes the main parts of the standalone application. We chose to make the downloading part via the ftp protocol. The wireless part is not encoded.

- The `FtpClient` class makes a connection to an existing ftp server and has a list of all distant images. It provides a `getSome` function to enable the downloading of many files at once.
- The `ImageBuffer` class is responsible for the management of downloaded files in a local directory. As

[1] http://trolltech.com/products/qt/

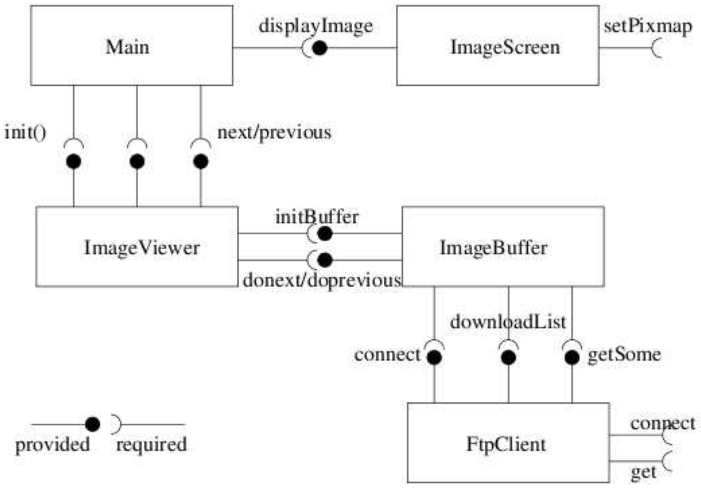


Fig. 3. Functional view of the application

the specification says, this buffer has a limited size and different policy for downloading images. The class provides the two functions donext and doprevious which are asynchronous functions. A signal is thrown if/when the desired image is ready to be displayed. It eventually downloads future images in current directory.

- The ImageViewer class is a high level component to make the interface between the ftp and buffer classes to the graphics components.
- The ImageScreen class is responsible for the display of the image in a graphic component named QPixmap.
- The main class provides all the graphics components for the Graphical User Interface.

### B. Integration of Qinna

Now that we have the functional part of the application, we add the following resource components: Memory, and Network which are QoSComponents that provide variable services. We only focus on these two basic resources. The Network component is only linked to the FtpClient, whereas Memory will be shared between all components. For Memory, the only variable service is amalloc which can fail if the global amount of dedicated memory is reached ; this function has only one implementation level. For Network, the provided function get can fail if there is too much activity on network (notion of bandwidth).

Then we follow the above methodology in the particular case of our remote viewer.

**Identification of the variable services (step 1)**

Now as the variable services for low level components have been identified, we list the following adaptive services for the functional part:

- ImageScreen.displayImage varies among memory, it has three implementation levels which correspond to the quality of the displayed image. We add calls to Memory.amalloc function to simulate the use of Memory.
- Ftpclient.getsome's implementation varies among available memory and the current bandwidth of network. If there is not enough memory or network, it adapts the policy of the downloads. It has three implementation levels. We add calls to Network.bandwidth to simulate the network resources that are needed to download files.
- ImageBuffer.donext/previous varies among available memory: if there is not enough memory the image is saved with high compression.

**Creation of the QoSComponents (step 2)**

The resource components are QoSComponents. Then, the three components ImageScreen, FtpClient and ImageBuffer are QoSComponents which provide each one variable service. Imageviewer and Main are QoSComponents as well. Figure 4 represents now the structure of the application at this step.

For the sake of simplicity, we only share Memory into two parts, a part for ImageBuffer and the other part for imageBuffer. That means that each of these components have their own amount of memory.

**Resource constraints (steps 3 and 4)**

The quantity of resource constraints we have fixed are classical ones (bounds for the memory instances, unique instantiation for the imageScreen component, no more than 80 percent of bandwidth for the ftpClient, etc). The QLSC are very similar to those described in [2] for a videogame application. Here we show how we have implemented some of these constraints in our application.

- *Quantity of resource constraints* The imageScreen component is responsible for the unique service display_image (display the image on the graphic video widget). Here are some behavioral constraints we implemented for this component:
  - There is only one instance of the component once.
  - The display function can only display images with size lesser or equal to $1200 * 800$.
  - There is only one call to the display function once.

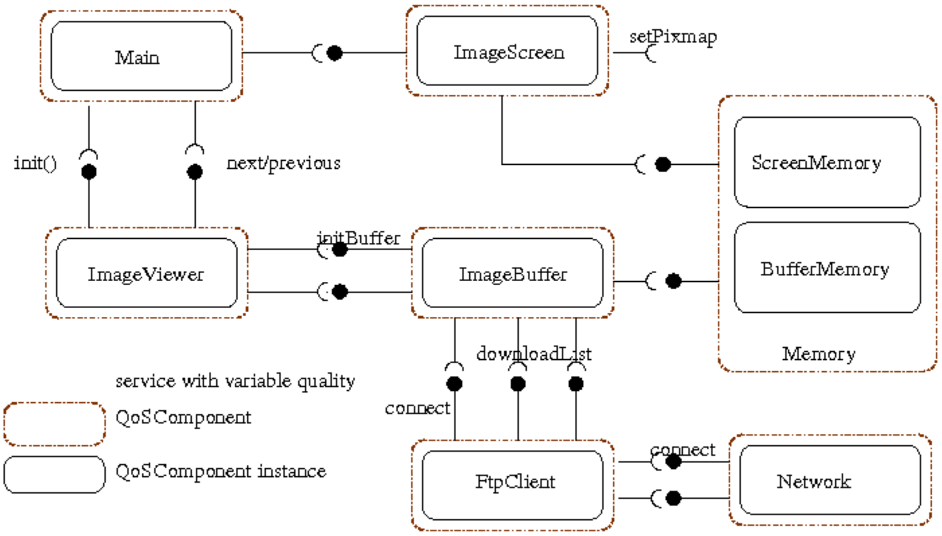


Fig. 4. Application with Qinna

These type constraints are easily implemented in the associated `imageScreenBroker` in the following way: the constraint "maximum of instance" requires two private attributes `nbinstance` and `nbinstancemax` which are declared and initialized at the creation of the Broker with values $0$ and $1$. Then the reservation of a new `imageScreen` by the Broker is done after checking whether or not $nbinstance + 1 \leq nbinstancemax$. If all checks are true, it reserves the instance and increments `nbinstance`.
The checking of memory is done by setting the global amount of memory for `ImageBuffer` and `imageBuffer` in local variables which are set to $0$ at the beginning of each contract, and updated each time the function `amalloc` is called.
These constraints are rather simple but we can imagine more complex ones, provided they can be checked with bounded complexity (this is a constraint coming from the fact the Qinna components will also be embedded).

- *QoS Linking constraints*
To illustrate the difference between quality of resource constraints and linking constraints, we show here the constraints for the `FtpClient.getSome`:
  - The implementation level $0$ corresponds to $3$ successive downloads with the `Network.get` function. The function has a unique implementation level but each call to it is made with $60$ as argument, to model the fact it requires $60\%$ of the total bandwidth. These three calls are made through the use of the `Thread.thread` with implementation level $0$ (quick thread, no active wait).
  - The implementation level $1$ corresponds to $2$ calls to the `get` function with $40\%$ of bandwidth each time. These two calls are made through the use of the `Thread.thread` with implementation level $1$ (middle thread, few active wait).
  - The implementation level $2$ corresponds to $1$ call to the `get` function with $20\%$ bandwidth. This call is made through the use of the `Thread.thread` with implementation level $2$ (more active wait).

Thus if the available bandwidth is too low, a negotiation or an existing contract will fail because of the resource constraints. The creation of the contract may fail because a thread cannot be provided at the desired implementation level.

**Modification of toplevel (step 5)** This part is straightforward. The only choices we have to make are the relative amount of resource (Memory, Network) which are allocated to each QoSComponents. The test scenario is detailed in section VI-D.

### C. Some statistics

The viewer is written in 4350 lines of code, the functional part taking roughly 1800 lines. The other lines are Qinna's generic components (1650 loc.), 600 lines of code for the new components (imagescreenBroker, imageScreenManager *etc.*) and 300 lines of code for the test scenarios. The binary is also much bigger  4.7Mbytes versus 2Mbytes without Qinna.

Thus Qinna is costly, but all the supplementary lines of code do not need to be rewritten, because:

- Generic Qinna components, algorithms, and the basic resource components are provided with Qinna.
- The decision functions for Quality of service constraints could be automatically generated or be provided as a "library of common constraints".
- The initialization at toplevel could be computed-aided through user-friendly tables.

We think that the cost of Qinna in terms of binary code can be strongly reduced by avoiding the existing redundancy in our current implementation.

Moreover, Qinna's implementation can be viewed as a prototype to evaluate the resource use and the quality of service management. After a preliminary phase with the whole implementation used to find the best linking constraints, we can imagine an optimized compilation through glue code which neither includes brokers nor managers.

### D. Results

We realized a scenario with a new component whose only objective is to use the basic resources Memory and Network. This `TestC` component provides only the `foobar` function at toplevel. This function has two implementation levels, and requires two functions: `ScreenMemory.amalloc` and `Network.get`. The whole application provides four functions at toplevel: `TestC.foobar`, `ImageViewer.donext` (and `doprevious`) and `ImageScreen.displayimage`. Three contracts are negotiated, in the following importance order: `foobar` first, then `donext` and `doprevious`, then `displayimage`. We made the three contracts and download and visualize images at the highest qualities, but at some point the foobar function causes the degradation of the contract for displayimage, and the images are then shown in a degraded version, like the Eiffel tower on Figure 2.

The gap between the characteristics of the contract and the effective resource usage can be make through the use of log functions provided by the Qinna implementation.

## VII. Related works

Other works also propose to use a development framework to handle resource variability. In [4] and [5], the author propose a model-based framework for developping self-adaptative programs. This approach uses high-level specifications based on temporal logic formula to generate program monitors. At runtime, these monitors catch the system events and activates the reconfiguration. This approach is similar to us except that it mainly deals with hybrid automata and there is no notion of contract degradation nor generic algorithm for negociation.

The expression and maintenance of resource constraints is also considered as a fundamental issue, so much work deals with this subject. In [6], the author use a probabilistic approach to evaluate the resource consumed by the program paths. Some other works in the domain of verification try to prove conformance of one program to some specification : in [7], for instance, the authors use synchronous observers to encode and verify logical time contracts. At last, the QML language ([8],[9]) is now well used to express QoS properties. This last approach is complementary to our one since it provides a language which could be compiled into source code for QoSComponents or Brokers.

## VIII. Conclusion and future work

In this paper, we have presented a case study using the software architecture Qinna which was designed to handle resource constraints during the development and the execution of embedded programs. We focused mainly on the development part, by giving a general development scheme to use Qinna, and illustrating it on a case study. The resulting application is a resource-aware application, whose resources constraints are guaranteed at runtime, and whose adaptation to variability of service is automatically done by the Qinna components, through the notion of contracts. At last, we are able to evaluate at runtime the threshold between contractualised resource and the real amount of resource effectively used.

This work has shown the effectivity of Qinna with respect to the programming effort, and the performance of the modified application.

Future work include some improvements of Qinna's C++ components, mainly on data structures, in order to decrease the global cost of Qinna in terms of binary size, and more specific and detailed resource components, in order to better fit to the platform specifications.

From the theoretical point of view, there is also a need for a way to manage the linking constraints. The developer has still to link the implementation levels of required and provided services, and the order between all implementations levels is fixed by him as well. The tuning of all these links can only be done though simulation yet. We think that some methods like controller synthesis ([10]) could be used to discover the/a optimal order and linking relations w.r.t. some constraints such as "minimal variability", "best reactivity" *etc.*.

Finally, some theoretical work would be necessary in order to use Qinna as a prediction tool, and provide an efficient compilation into "glue code".

## References

[1] M. Sparling, "Lessons learned through six years of component-based development," *Commun. ACM*, vol. 43, no. 10, 2000.

[2] J.-C. Tournier, "Qinna: une architecture à base de composants pour la gestion de la qualité de service dans les systèmes embarqués mobiles," Ph.D. dissertation, INSA-Lyon, 2005.

[3] J.-C. Tournier, V. Olive, and J.-P. Babau, "Towards a dynamic management of QoS constraints in embedded systems," in *Workshop QoSCBSE, in conjunction with ADA'03*, Toulouse, France, June 2003.

[4] L. Tan, "Model-based self-monitoring embedded systems with temporal logic specifications," in *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering (ASE'05)*, 2005.

[5] I. Lee, S. Kannan, M. Kim, O. Sokolsky, and M. Viswanathan, "Runtime assurance based on formal specifications," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (IPDPS'99)*, 1999.

[6] H. Koziolek and V. Firus, "Parametric Performance Contracts: Non-Markovian Loop Modelling and an Experimental Evaluation," in *Formal Foundations of Embedded Software and Component-Based Software Architectures (FESCA)*, ser. Electronical Notes in Computer Science, Vienna, Austria, 2006.

[7] F. Maraninchi and L. Morel, "Logical-time contracts for reactive embedded components," in *30th EUROMICRO Conference on Component-Based Software Engineering Track, ECBSE'04*, Rennes, France, Aug. 2004.

[8] S. Frølund and J. Koistinen, "Quality of services specification in distributed object systems design," in *Proceedings of the 4th conference on USENIX Conference on Object-Oriented Technologies and Systems (COOTS)*. Berkeley, CA, USA: USENIX Association, 1998.

[9] ——, "Qml : A language for quality of service specification," HPL-98-10, Tech. Rep., 1998.

[10] F. M. K. Altisen, A. Clodic and E. Rutten, "Using controller synthesis to build property-enforcing layers," in *European Symposium on Programming (ESOP)*, April 2003.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 625 – 628

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Real-time Control Teaching Using LEGO® MINDSTORMS® NXT Robot

Wojciech Grega,
AGH University of Science and Technology,
Department of Automatics, 30-059 Krakow, Poland,
Email: wgr@ia.agh.edu.pl

Adam Piłat
AGH University of Science and Technology,
Department of Automatics, 30-059 Krakow, Poland,
Email: ap@ia.agh.edu.pl

***Abstract*—A current trend in learning programs, even at high-degree studies, is applying the concepts of "learning by projects". In this context, the LEGO® MINDSTORMS® NXT modular robot appears as a simple, flexible and attractive educational platform to achieve the referred challenge in many domains of information technologies, especially in real-time control. The project team can operate with the real system (e.g. constructed robot) to realize a number of particular tasks. The ready-to-use embedded platform and dual microcontrollers architecture represents actual trends in the modern control systems designs. The open hardware and software architecture gives unlimited possibility to handle all devices connected to the main control brick and write effective control algorithms.**

**The proposed rescue robot project is a good example of the simple system where, a number of real-time control problems can be analyzed. The project on this scale requires detailed planning, cooperation in teams, extensive literature survey, and comprehensive software design. That is exactly, what we need for "learning by projects" concept.**

## I. Introduction

A current trend in learning curricula, especially at engineering studies, is applying the concepts of "learning by experiments" or "learning by projects" [1]. The method gives the students the opportunity to get familiar with some practical problems that faces project development teams and organizations in real situations. It was found, that learning achievements resulting from the application of a collaborative work methodology based on the "learning by projects" are much higher [2].

This concept was also proposed for ILERT (International Learning Environment for Real-Time Software Intensive Control System) EU/US project [3]. This study leads to establishing a methodology for a multinational, engineering program, producing graduates capable of working efficiently in multidisciplinary teams engaged in international collaboration on industrial projects. One important output the proposed study is creation of an interdisciplinary specialization in Real-Time Software-Intensive Control (RSIC). As a part of the pilot implementation phase of ILERT project selected courses were introduced as a RSIC curriculum units, acceptable for engineering programs in four partner organizations.

During the research phase of the ILERT we did an intensive literature study demonstrating, that several technical universities have practically implemented robotic design and control experiments with the LEGO MINDSTORMS kits [4], [5]. The LEGO kits give the students a rich and flexible material, which they can use in their design of their robotics projects. The LEGO ability to link directly with Simulink® and access the MATLAB® toolboxes has classified this robot kits as an "open" laboratory. The LEGO MINDSTORMS NXT design is based on the advanced 32-bit ARM7 microcontroller, which can be programmed with the LabView based block-oriented language. However, many other programming tools and languages applicable for real-time LEGO robot control experiments are available in the Internet. The published hardware [9] and software [10] documentation makes the LEGO® MINDSTORMS® NXT system open for any kind of modifications and new applications. Several authors are replacing the operating system of the MINDSTORMS microcontroller with a real-time operating system suitable for C/C++ or similar applications. Most popular is NXC language [17], which supports all of the commands provided by the ARM7 microcontroller. This upgrade might be advantageous for some real-time control applications. The "upgraded" LEGO MINDSTORMS experiment can teach students the importance of real-time computing, periodic tasks development, timers and tradeoffs involving embedded processor size and cost versus performance relations within the context of control systems.

## II. Teaching requirements

From educational point of view, if control aspects are the key element of the curricula, student are expected to design control algorithms based on the mathematical model, simulate how the controlled system works and then implement the controller for the robot during the limited time of a regular academic course. In this case integration of real-time environment with rapid prototyping platforms such as MATLAB/ Simulink or LabView is essential.

If other topics of real-time control system must be covered, for example, task scheduling, resource management, real-time communication or fault-tolerance then text based or object oriented languages are preferable.

This work was supported by the AGH research grant

It is also important to demonstrate  to the students how to fulfill the needs for embedded systems, e.g. how to minimize the application memory requirements. It will be also valuable, if the real-time system monitoring tools provide on-line facilities for measuring execution times.

Working with communication algorithms the LEGO MINDSTORMS built-in Bluetooth media or IEEE 802.11 extension can be explored. The communication tasks can be focused on information interchange between two or more robots and host PC. The PC can support a higher level control algorithms or just to be used for monitoring and data acquisition purposes.

## III. Real-time operating systems for LEGO® MINDSTORMS® NXT

The NXT-G standard graphical programming environment for LMNXT is useful only for very beginning  experiments with robots. Its limited performance does not allow to create and diagnose in real-time the specific algorithms. To improve the performance of robot control a number of software tools were developed and are available [8]. The programming environments available for the LEGO® MINDSTORMS NXT robot and representing real-time features are listed in Table 1. Two most popular solutions for academia and teaching purposes are: LEGO MINDSTORMS Toolkit for LabView and Embedded Coder Robot NXT - based on the graphical environment. Both of them need the licensed software and a number of toolboxes installed. The LabView can operate with a standard LEGO firmware while MATLAB requires the replacement by the leJOS system.

Text programming solutions are represented by NXC and leJOS – both of them are free and open source solutions. The NXC (Not Exactly C) is a text language for MINDSTORMS® NXT robot microcontroller programming with multitasking features, based on standard LEGO firmware [6].

With free programming environment IDE [17] it is possible to create complex and advanced data acquisition and control algorithms operating in real-time, including file management and Bluetooth communication features implementation.

Using text language the programming skills of students together with their understanding of data precision, time dependencies and I/O devices access methods to can be improved.

The leJOS is a Java-based replacement firmware for the LEGO MINDSTORMS RCX microcontroller [14]. The leJOS environment is based on object oriented language (Java) and has the following features: preemptive threads (tasks), arrays including multi-dimensional, recursion, synchronization, exceptions. Java variable types includes Float, Long, and String.

The nxtOSEK [13] (previous name – up to June 2008 was leJOS OSEK) is a hybrid of existing two open source projects (leJOS NXJ and TOPPERS OSEK):

- leJOS NXJ is a API device for NXT sensors, motors, and other external devices [14],
- TOPPERS OSEK provides real-time multitasking features proven in automotive industry [16]. OSEK was originally designed for embedded real-time control systems which are used in real cars [15].

The nxtOSEK is focused on real-time control applications for LEGO MINDSTORMS RCX, thus user-friendly GUI/file system was out of target. Additionally, one can use a graphical modeling, simulation, and code generation environment which is called Embedded Coder Robot NXT - specific MATLAB/Simulink Toolbox.

A unique feature of nxtOSEK application is that users do not need to apply time wait API, which is frequently used by other NXT programming languages to execute control algorithm at desired timing. The nxtOSEK provides accurate preemptive and periodical/event driven task scheduling.

An interesting programming interface, based on client/server architecture , is URBI (Universal Real-time Behavior Interface) [7]. This scripting language can be interfaced with several popular programming languages (C++, Java, MATLAB,...) and OS (Windows, Mac OSX, Linux).

Table I.
Selected programming tools for LEGO MINDSTORMS NXT

| | **LabVIEW Toolkit** | **Matlab/ Simulink** | **NXC** | **RobotC** | **leJOS NXJ** | **nxtOSEK** | **URBI** |
|---|---|---|---|---|---|---|---|
| Programming | Graph | Graph | Text | Text | Text | Text | Text |
| Syntax | NI blocks | Simulink blocks, C | Like C | C | Java | C/C++ | Like C/C++ |
| Firmware | Standard | Repl. | Standard | Repl. | Repl. | Repl. | Standard |
| License | LabView | Matlab/Simulink | Freeware | Yes | Open source | Open source | Open source |
| Events | No | Yes | No | Yes | Java events | Yes (OSEK RTOS) | Yes |
| Multithreading | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Bluetooth communication: Brick to PC | Yes | Yes | Yes | Yes | Yes | Yes | n.a. |
| BluetoothBrick to Brick | Yes | Yes | Yes | Yes | Yes | Not yet | n.a. |
| Floating point | No | No | No | Yes | Yes | Yes | Yes |

The URBI language uses an C++ like syntax. It allows to obtain parallel execution of commands, event programming, command tagging, dynamic variables. With the event feature a user can react on sensor inputs or create individual events and emit them ones or periodically.

## IV. Example: NXC – based system for robot DC motor control

The demo LEGO MINDSTORMS NXT robot has been developed for RSIC teaching purposes. The robot uses two drives for movement and one for area scanning with distance and light sensor.

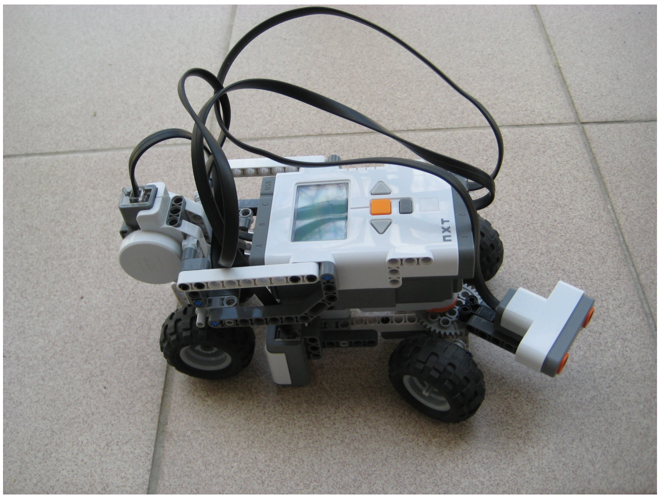


Fig 1. Sinusoid

The problem of target search in the maze was defined as a practical example (“rescue problem”). One can imagine a few algorithms  solving this problem. The basic problem of the proposed robot is  accurate and precise motor control. The error in the motor control can affect the correct navigation and robot movements. From the practical point of view, it is not so important for rescue operation in the maze that the robot moves, but how precise it operates and what is the accuracy of the controlled devices. Thus, the tasks execution details and precision of the robot controller are important.

The aim of this part of the project was to analyze the motor control using two tasks dedicated to motor PID controller and motor encoder measurements. Before the any multitasking operation is started it is required to initialize semaphores, files, read timer values and start simultaneous task execution. The aims of these two tasks were as follows:

- TASK 1 – “Motor control”: to read Tick Counter and Log data to file, start Motor PID control, set semaphore when PID control loop terminates, read Tick Counter and Log data to file,
- TASK 2 - “Encoder Means”: to read Tick Counter and Log data to file, read motor rotations and speed and log them to file, till semaphore is inactive, read Tick Counter and Log data to file (Fig.2).

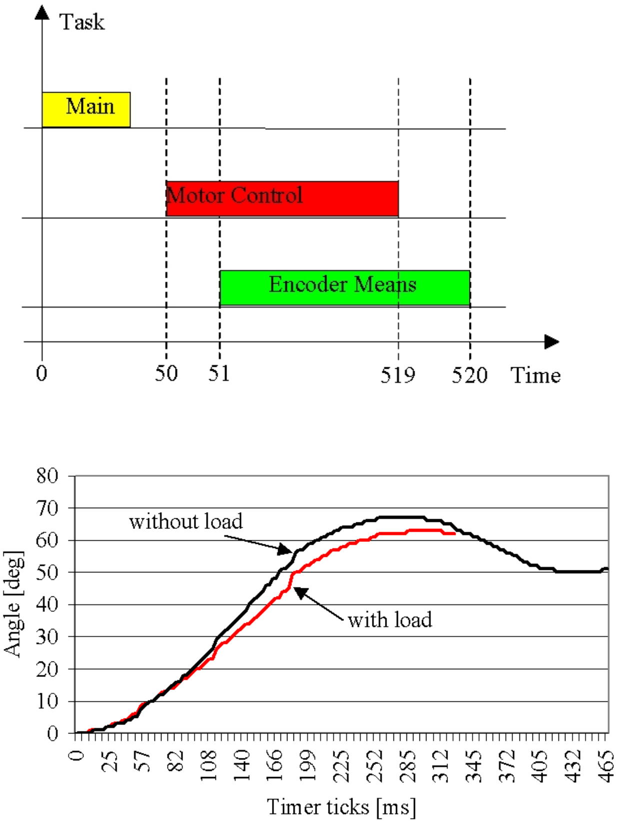


Fig. 2. Motor control in real-time with dual tasks: a) task schedule, b) measured rotor position

The control task was to set the desired angle of the motor axis (45 degrees) using 75% of the total motor power and setting the PID controller parameters. The realized experiments and logged data with a time resolution of 1ms show how the particular tasks are executed. The tasks time diagrams (Fig. 2a) show the execution sequence and confirm the data log time slots. The motor position time diagrams demonstrate the difference in the PID control quality due to the motor load. The steady-state error shows the regulation mismatch and the length of the logged data sequence points the time when the control task has been finished.

The realized testing program written in NXC shows its possibilities for real-time control education. Working with motor control problem students can analyze performance of the built-in, firmware digital PID controller or prepare a custom version of control algorithm. Running two (or more) tasks in the concurrent mode and logging time slots, the scheduling and tasks execution subroutines can be traced. The running tasks can be analyzed in the case of start-up, progress and terminate action.

## V. Conclusions

The idea of open hardware and software solutions for LEGO® MINDSTORMS® NXT offers an unlimited number of application in the field of control education. If focusing of the real-time aspects, the text-based languages are required to demonstrate a full control over the executed code.

Analyzing several solutions it seems that the NXC and nxtOSEK are the best candidates to support real-time control experiments. The simplicity of NXC allows to start immediately with programming, while to start nxtOSEK a number of software replacements is required.

At this moment there are some limitations of Bluetooth communication using nxtOSEK, but hopefully this task will be extended soon. The presented simple example shows that it not so easy to obtain a precise and perfect results of the low level task. One can enjoy with working robot, but the important question facing the students is: does we have a full control providing a repetitive behavior of the robot?

## Acknowledgment

The authors gratefully acknowledge the initial work done by their students K. Balwierz, D. Bieniek J. Blicharski and Ł. Łucek during the first stage of the project.

## References

[1] Bassam A. Hussein, Kjetil Nyseth , "A method for learning in project management, 'Learning by projects' ", *9th International Workshop on Experimental Interactive Learning in Industrial Management,* "New Approaches on Learning, Studying and Teaching", Espoo, Finland, June 5-7, 2005

[2] Grega W., Piłat A., "Platforms for Laboratory Experiments: Low-cost Kits vs. Dedicated Trainer"s, *Proceedings of 18th EAEEIE Conference,* Praha July 2-4, ISBN 978-80-01-03745-4

[3] Grega W., Kornecki A., Sveda M., and Thiriet J.M., "Developing Interdisciplinary and Multinational Software Engineering Curriculum", *Proceedings of the ICEE'07,* Coimbra, Portugal, Sep. 3-7, 2007

[4] Gawthrop P. J., McGookin E., "A LEGO®-Based Control Experiment", *IEEE Control Systems Magazine,* v.24, October 2004, pp. 43-56

[5] Wang E. L., LaCombe J., and Rogers C., "Using LEGO® Bricks to Conduct Engineering Experiments," *Proceedings of the ASEE Annual conference and exhibition,* session 2756, 2004

[6] LEGO firmware, http://mindstorms.lego.com

[7] URBI, www.urbiforge.com, 2008

[8] http://www.teamhassenplug.org/NXT/NXTSoftware.html, 2008

[9] LEGO® MINDSTORMS® Hardware Developer Kit (HDK), http://mindstorms.lego.com, LEGO Group, 2006

[10] LEGO® MINDSTORMS® Software Developer Kit (HDK), http://mindstorms.lego.com, LEGO Group 2006

[11] Embedded Coder Robot NXT Demo, http://www.mathworks.com, 2006

[12] NXTway-GS (Self-Balancing Two-Wheeled Robot) Controller Design, http://www.mathworks.com, 2008

[13] nxtOSEK, http://lejos-osek.sourceforge.net, 2008

[14] leJOS, http://lejos.sourceforge.net, 2008

[15] OSEK, http://portal.osek-vdx.org, 1993

[16] TOPPERS Project, http://www.toppers.jp/en/index.html, 2003

[17] Hansen J., "Not eXactly C (NXC) Programmer's Guide", 2007

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 629–632

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Safety Shell for UML-RT Projects

Roman Gumzej
University of Maribor
Faculty of Electrical Engineering and Computer Science
2000 Maribor, Slovenia
*roman.gumzej@uni-mb.si*

Wolfgang A. Halang
Fernuniversität
Chair of Computer Engineering and Real-time Systems
58084 Hagen, Germany
*wolfgang.halang@fernuni-hagen.de*

***Abstract*—A safety shell pattern was defined based on a reconfiguration management pattern, and inspired by the architectural specifications in Specification PEARL. It is meant to be used for real-time applications to be developed with UML-RT as described. The implementation of the safety shell features as defined by in [8], namely its timing and state guards as well as I/O protection and exception handling mechanisms, is explained. The pattern is parameterised by defining the properties of its components as well as by defining the mapping between software and hardware architectures. Initial and alternative execution scenarios as well as the method for switching between them are defined. The goal pursued with the safety shell is to obtain clearly specified operation scenarios with well defined transitions between them. To achieve safe and timely operation, the pattern must provide safety shell mechanisms for an application designed, i.e., enable its predictable deterministic and temporally predictable operation now and in the future.**

## I. Introduction

PREDICTABILITY, dependability and timeliness are major pre-conditions for real-time applications to be safe. Hence, in order to be able to effectively address safety, it should be sustained throughout the entire life-cycle of an application – from design via implementation to upgrades and maintenance. Therefore, it appeared sensible to build a design pattern that would enable designers to address most safety issues and build safe and persistent applications. Kornecky and Zalewski [8] have defined a "safety shell" for real-time applications to be composed of several "guards", each one protecting a certain part of an application. Thus, the input/output needs to be protected from tampering as well as by range checking to sustain the environmental parameters of the application. Then, exception-handling mechanisms should protect the application from malicious consequences of unforeseen situations, by offering mechanisms that bring it back to normal operation. Finally, the operation should be monitored and safeguarded in its state and time spaces in order to prevent the application from leaving its specified execution and temporal frameworks. Since all mechanisms mentioned foresee different scenarios for initialisation, phases of normal operation and of various exception modes, enabling dynamic re-configurations on the application level is crucial to enable these features.

In the development of embedded real-time systems, management of dynamic (re-) configuration has systematically been addressed by hardware/software co-design methods (cp., e.g., [9, 11, 13]). Besides defining diverse (dynamic) operation scenarios, two main goals have been targeted by this approach:

1) achieving fault tolerance by system design (cp. [3, 6]),
2) fast scenario switching (e.g., in industrial automation and telecommunication systems [1, 4, 5, 11]).

## II. Design for Safety

When designing a real-time system, generally three viewpoints must be considered:

1) the external (functional) one, which represents the inputs/outputs and usage scenarios,
2) the internal (behavioural) one, which deals with the definition of usage scenarios, and
3) the definition of system structure – hardware and software architectures together with the mapping of software components onto hardware components and the definition of configurations and re-configuration scenarios.

During re-configuration, application data must remain consistent and real-time constraints must be satisfied. In order to be able to achieve this, these issues must be addressed at multiple levels. At the lowest level, the hardware must be reconfigurable. Software-programmable hardware components support this inherently, since their functions can be changed by their memory contents. Internal hardware structures are designed to restrict dangerous conditions that could damage hardware. At the next higher level, the internal states of the software must be managed under changing tasking. Operating systems support flexible implementations of multiple tasks on single processors in form of time-sharing and/or multitasking. On the top level, one wants to define operation scenarios – configurations – for an application, which enable it to adapt to varying conditions in the environment on one hand, and to respond to changing operational modes by switching between operation scenarios in a safe and predictable manner on the other. Typically, these configurations cannot be managed by operating systems, since groups of processes and possibly also hardware components are involved. Hence, their management is usually placed on the application or middleware level, since it requires the observation of and actions based on the system state. Generally, by this approach, low-level efficiency and hard real-time properties are difficult to achieve. Because of this, we chose to distribute re-configuration management to all three levels – hardware, middleware, and software. With this in

978-83-60810-14-9/08/$25.00 © 2008 IEEE

mind, the hardware/software co-design profile and pattern for real-time application design in UML based on the specification language Specification PEARL (S-PEARL) (cp. [2]) have been developed. While in the profile the constructs of S-PEARL are introduced with their properties, behaviour and interconnections, the configuration management pattern provides the mapping of software to hardware components, and a foundation on which to build custom real-time applications. The latter's approach is followed in extending and parameterising the configuration management pattern with safety features. The pattern and its safety-oriented use are presented throughout this article with the goal to construct a safety shell (cp. [7, 8]) for a designed application.

### A. Configuration Management Pattern

The configuration management pattern was constructed by combining a set of UML-RT [12] stereotypes [10], which represent S-PEARL constructs (see Fig. 1) as a coherent whole. They constitute building blocks at the three levels of architectural modeling: (1) hardware architecture, (2) software architecture, and (3) software-to-hardware mapping. A hardware architecture is composed of processing nodes, termed "stations", whose descriptions also mention their components with their properties. A software architecture is organised in the form of "collections" of "modules", comprising program "tasks", functions and procedures of the application software. The "collection" is the unit of software to be mapped onto "stations", i.e., at any time there is exactly one collection assigned to run on a station. Thus, the "collection" is also the unit of dynamic re-configuration. Each of these constructs has its specific attributes, which pertain to all objects of this type (such as properties, relations, and initialisation). They are layered on three levels of abstraction (see Fig. 2):

1) station level, where a mapping of collection configurations to stations is established;
2) configuration level, where collections grouped into scenarios, named "configurations", are managed by a "reconfiguration manager"; and
3) collection level, where the tasks, which may be grouped into modules (UML packages), are managed by their collections according to their scheduling parameters.

Each collection belongs to a configuration and is mapped to a station. Configuration management is responsible for the co-operation among collections and possible dynamical re-configurations, which depend on the state changes of the station they are residing on. A detailed description of its safety-oriented use is presented in the sequel.

## III. Introducing a Safety Shell

A safety shell is responsible for guarding the main process termed "Primary control" (see Fig. 3) by providing it with additional functionality which keeps the possible sources of error to a minimum. In order to be effective, these functions have to be integrated into or built around an application. In our case, the second variant was chosen by implementing a pattern, which forms the "backbone" of an application, requiring it to be formed in a specific manner in order to function in the safety shell's environment. The configuration management pattern has the structure and functions needed to fulfill the rôle of a safety shell in terms of guarding a system in such a way that it always remains in a foreseen state and time frame of operation. In the sequel, the functions of the four protective mechanisms are described, and it is explained in which manner and to which degree they safeguard an application's execution.

| S-PEARL Element | UML Element | Stereotype | Icon |
|---|---|---|---|
| Station | Capsule | <<SPStation>> | ▭ |
| Workstore,... | Class | <<SPWorkstore>>,... | ▭ Workstore ▬ Device ⬭ Proctype |
| Line | Class | <<SPLine>> | — |
| Collection | Capsule | <<SPCollection>> | ▭ |
| Port | Class | <<SPPort>> | |
| Module | Package | | ⊟ |
| Task | Capsule | <<SPTask>> | ⬯ |

Fig. 1. S-PEARL constructs and their UML (-RT) stereotypes

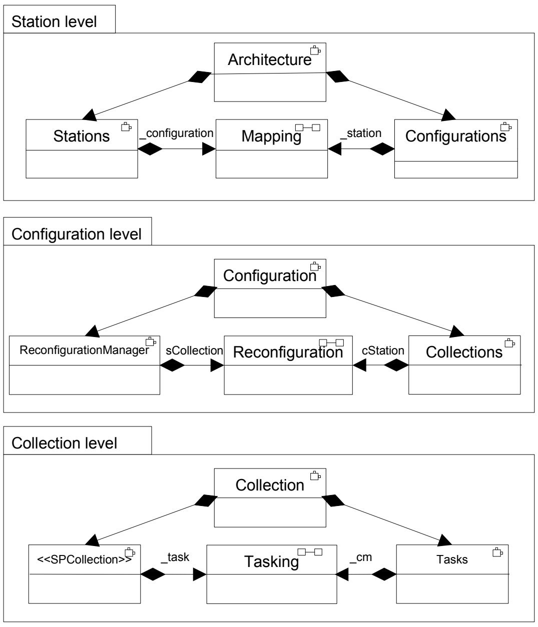


Fig. 2. Levels of configuration management

### A. Protected Input/Output

Protected input/output refers to well defined interfaces with the environment. By well defined we mean stable physical connections and sound protocols with integrated error checking and correction techniques. Usually, the possible problems originate from data overruns or malicious data. By themselves, the device drivers of interfaces can only correct a part of these problems associated with data formats and protocols. They could, however, also detect overruns or out-of-scope data, prevent recurring corruption of data, and signal possible errors. In our implementation of I/O *ports* (see Fig. 4), an important

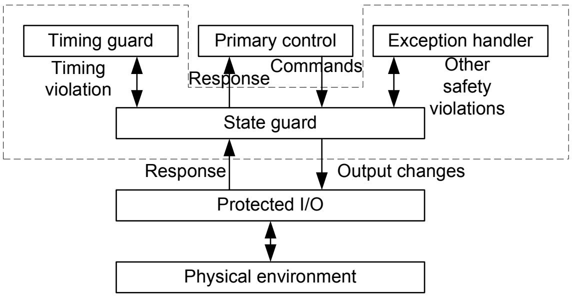


Fig. 3. Safety shell scheme

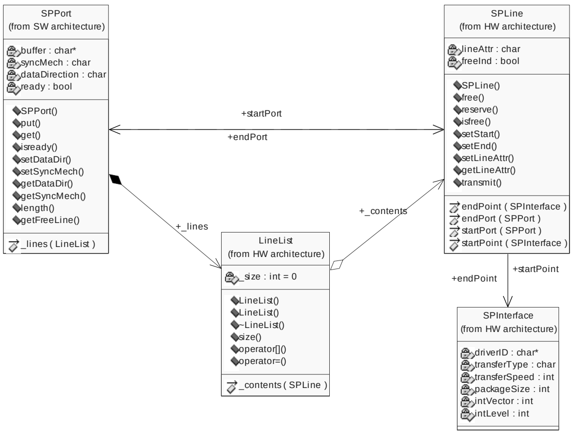


Fig. 4. Safeguarded port-to-port communications

safety-related feature is present, namely, routing parameters. Where stable line connections are of utmost importance, they are usually designed redundantly, being doubled, tripled, or with one of the *lines* representing a slower yet reliable (e.g., wireless) connection. In our routing parameters we can determine the lines which can/must be used, and/or assign a preferred line, being the fastest or most trusted one. If a line is not or becomes unavailable, the protocol automatically searches for the next available line. On the application level, it is important to have uniform interfaces between software components possibly executing at different processing nodes, which is also achieved by "port-to-port" communication. The lower levels are suppressed on the application level, however the parameterisation of the connection lines between ports and device drivers is made transparent by the S-PEARL profile, and enables complete oversight down to the physical level.

### B. State Guard

There should be a predefined scenario for each possible state a system can find itself in ("state guard"). The problem decomposition enables considering loosely coupled interdependent processing nodes, which ensure local predictability, and have well defined global interconnections that form an integral part of each scenario. Due to a possible state explosion during execution, it is impossible to (pre-) determine all global states a system can assume and define scenarios for them. Since scenarios are defined for each station separately (e.g., see Fig. 5), this problem becomes easier to tackle. There exists a limited number of states, only, and as global interconnections are (re-) connected during re-configuration, the global implications are unproblematic. Hence, besides *fault containment*, this makes designing distributed real-time applications easier, and the systems designed more robust. A rigorously designed application structure, in which the execution of a collection of tasks is associated with an exactly pre-determined state, and a simple and well defined station state changing mechanism prevent the transition to an undefined state, herewith implementing the state guard function. The same holds for tasks, where an exactly defined activity structure does not only prevent transitions to an undefined state by exception handling, but also supports safe-guarded execution of temporally sensitive operations (see Fig. 6).

```
// "initial_load_statement"
void init() {
  reconfigure(0); // '0' is the initial state
}

// "reconfiguration_statement"
void reconfigure(char s) {
  if (sreg!=s) { // sreg is the station state register
    switch (sreg) {
      case -1: // initially it is undefined
        break;
      default: // upon state change the current collection is unloaded
        sCollection.unload().sendAt(sreg);
        break;
    }
    switch (s) {
      case -1: // if there is no valid state, nothing is loaded
        break;
      default: // the collection associated with the new state is loaded
        sCollection.load().sendAt(s);
        break;
    }
  }
  sreg=s;
}
```

Fig. 5. Safeguarded state transitions within a station (configuration)

### C. Timing Guard

Timeliness, being a critical property of real-time systems, is of utmost importance for applications and, hence, it is vital that its "backbone" does not introduce any significant delays into execution. Due to this and to ensure observability, the service algorithms of the pattern have been kept simple. They introduce no unbounded delays into scenario switching. This was one of the pre-dispositions while designing them, and is as important for safety as for timeliness of operation. Since some operations such as scheduling, message transmission, or task activation still require some time, however, the operating system overhead shall not introduce any unbounded delays either and, moreover, the service times of operating system calls have to be incorporated into task/operation execution times. Hence, the execution of an underlying real-time operating system has also to be temporally predictable in order to enable timeliness. In our implementation this was accomplished by a small custom real-time operating system (RTOS) kernel, which could be substituted with a fully functional off-the-shelf RTOS, keeping in mind the restriction on bounded and

```
void taskmain()
{
      if (_start_state) {
            aState=choose_start();
            _start_state=false;
            _timeout=false;
      }
      switch (aState) {
            case ith_state: {
                  if (_timeout) {
                        /* timeout action */
                  }
                  else {
                        _timeout=true; /* watchdog start*/
                        maxT.informIn(RTTimespec(timeout,0));
                        /* activity */
                        aState= ith_next();
                        maxT.cancelTimer();
                        _timeout=false; /* watchdog end*/
                  }
                  break;
            }
            case end_state: {
                  /* activity*/
                  _start_state=true;
                  _timeout=false;
                  break;
            }
            otherwise:
                  break;
      }
}
```

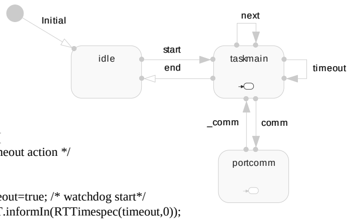


Fig. 6. Temporally safeguarded execution of task activities

predictable system call service times. Temporal monitoring of atomic activities such as executions of task operations is possible by introducing timers into them and, in this sense, prohibit any unreasonably long executions of atomic activities (e.g., by using a watchdog mechanism – "time guard" – see Fig. 6). Here it is also possible (for any activity – task state) to define a time-out action, which is executed in response to a possible time-out condition.

### D. Exception Handler

Since tasks can be re-scheduled at pre-emption points, only, and tasking operations are defined for active tasks, it is sensible to limit their duration. In case of violating an execution time frame, a pre-defined scenario could be activated representing, for instance, *graceful degradation*. To further support *fault-tolerant* operation, however, it would be desirable to additionally check the correctness of other vital operation parameters, too, and by doing so implementing other features of the – "exception handler" – as well. This may introduce additional overhead, but as long as the execution times remain predictable and in the foreseen time frames, this is not a problem. Exception handling is, in part, already present in the safeguarded I/O operations. As already mentioned, the port-to-port communication protocol also enables line replication, thus allowing for "no single point of failure" planning. Range and other error checking mechanisms could be implemented to ensure *fail-safe* operation using well known mechanisms in the same manner as we implemented the time-out handling by introducing, e.g., pre-/post-condition checking of the (critical) tasks' activities. To support *reversion modes*, several collections with the same functionality may be defined within the same configuration to be activated depending on the different operational modes. The "context" of a configuration could be maintained as a list of "collection contexts" or – as in our case – in the form of procedures to (re-) establish collections. They can be (re-) established while (re-) loading and (re-) connecting their ports when desired/needed. The choice to (re-) start or continue a collection execution depends on the nature of the application and is, hence, left to the designer. Although typically one would continue from a state-switching condition, it is not always desirable or even dangerous to do so. In most cases, a collection is only re-loaded if its initial pre-conditions and environment state have been re-established. The "collection context" itself consists of its task control block (TCB) table as well as the lists of currently active tasks and ports. All of these would need to be re-established when re-loading a collection for execution continuation.

## IV. Conclusion

With the help of the safety shell features of the S-PEARL configuration management pattern presented, distributed real-time application programs designed with UML-RT can run with safe, predictable behaviour and re-configuration support. Besides the structure of the applications, the configuration management pattern also defines uniform interfaces and protocols for intra- and inter-component/-node communication using pre-defined port/interface definitions. In (hard) real-time systems, it shall provide the necessary support for deterministic and dependable dynamic system re-configuration. The safety shell features are enabled by the pattern, but the selection and usage of the mentioned mechanisms remain the responsibility of a real-time application's designer, since no two safety critical real-time applications are equal.

## References

[1] Eisenring M., Platzner M. and Thiele L.: Communication Synthesis for Reconfigurable Embedded Systems. P. Lysaght, J. Irvine, R. W. Hartenstein (Eds.): *Field-Programmable Logic and Applications, Proc.* (1999) 205–214, Berlin: Springer-Verlag.

[2] Gumzej R., Colnarič M. and Halang W. A.: A Reconfiguration Pattern for Distributed Embedded Systems. *Software and Systems Modeling* (2007) Springer-Verlag. http://dx.doi.org/10.1007/s10270-007-0075-7

[3] Hofmeister C. R.: Dynamic Reconfiguration of Distributed Applications. *PhD thesis* (1993) University of Maryland.

[4] Hutchings B. L. and Wirthlin M. J.: Implementation Approaches for Reconfigurable Logic Applications. *Field-Programmable Logic and Applications, Proc.* (1995) 419–428, Berlin: Springer-Verlag.

[5] Jean J., Tomko K., Yavgal V., Cook R. and Shah J.: Dynamic Reconfiguration to Support Concurrent Applications. *IEEE Symposium on FPGAs for Custom Computing Machines, Proc.* (1998) 302–303, Los Alamitos: IEEE Computer Society Press.

[6] Kalbarczyk Z. T., Iyer R. K., Bagchi S. and Whisnant K.: Chameleon: A Software Infrastructure for Adaptive Fault Tolerance. *IEEE Trans. Parallel and Distributed Systems 10(6)* (1999).

[7] Katwijk J. van, Toetenel H., Sahraoui A., Anderson E. and Zalewski J.: Specification and Verification of a Safety Shell with Statecharts and Extended Timed Graphs. *Computer Safety, Reliability and Security* (2000) 37–52, LNCS 1943, Berlin: Springer-Verlag.

[8] Kornecki A. J. and Zalewski J.: Software Development for Real-Time Safety – Critical Applications. *Software Engineering Workshop – Tutorial Notes, 29th Annual IEEE/NASA 03* (2005) 1–95.

[9] Kramer J. and Magee J.: Dynamic Configuration for Distributed Systems. *IEEE Trans. Software Engineering 11(4)* (1985).

[10] Object Management Group: Unified Modeling Language: Superstructure. Version 2.0. *OMG document formal/2005-07-04* (2005).

[11] Rust C., Stappert F. and Bernhardi-Grisson R.: Petri Net Design of Reconfigurable Embedded Real-Time Systems. *IFIP 17th World Computer Congress – Design and Analysis of Distributed Embedded Systems, Proc.* (2002) 41–50, Dordrecht: Kluwer.

[12] Selić B. and Rumbaugh J.: Using UML for Modeling Complex Real-Time Systems. *Rational Software Corporation, White Paper* (1998) http://www.rational.com/media/whitepapers/umlrt.pdf

[13] Wolf W.: A Decade of Hardware/Software Codesign. *IEEE Computer 36(4)* (2003).

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 633 – 638

ISBN 978-83-60810-14-9
ISSN 1896-7094

# An RSIC-SE2004 Curriculum Framework

Thomas B. Hilburn
Embry-Riddle Aeronautical University, U.S.A., Email: hilburn@erau.edu

Andrew Kornecki
Embry-Riddle Aeronautical University, U.S.A., Email: kornecka@erau.edu

Jean-Marc Thiriet
Grenoble Université, France, Email: jean-marc.thiriet@ujf-grenoble.fr

Wojciech Grega
AGH University of Science & Technology, Poland, Email: wgr@agh.edu.pl

Miroslav Sveda
Brno University of Technology, Czech Republic, Email: sveda@fit.vutbr.cz

*Abstract* — **This paper addresses the problem of educating software engineering professionals who can efficiently and effectively develop real-time software intensive control systems in the global community. A framework for developing curricula that support such education is presented. The curriculum framework is based on the work of two education projects: the ILERT (International Learning Environment for Real-Time Software Intensive Control System) project, and the software engineering efforts of the ACM/IEEE-CS Joint Task Force on Computing Curricula. The authors describe a curriculum framework that integrates principles, content, and organization from the two projects, and which satisfies the intent and requirement of both projects.**

## I. Introduction

THE development of software for real-time, embedded, safety-critical systems (such as for controlling and supporting systems in aviation and space, medicine and health, and atomic energy) is a complex and challenging problem. The health, safety, welfare, and productivity of the public and the world economy increasingly depend on software-intensive systems – the dependability of these systems is paramount. Unfortunately, software developers do not consistently provide safe, secure, and reliable systems; such systems are often of poor quality, and cost and schedule overruns are common. There has been significant improvement in software development methods and practices in the last twenty years. The application of software engineering best practices to such development has proven the value of such practices [1]-[3]. In order to enhance such capabilities for software engineering professionals there have been numerous calls to improve software engineering education, especially in the area of real-time embedded systems [4]-[6].

The last twenty years have also witnessed significant advancements in the state of computer science education (and in allied fields such as computer engineering, information systems, and software engineering). The Association for Computing Machinery (ACM), the IEEE Computer Society (IEEE-CS), and CSAB (which determines criteria for accreditation of programs in computer science and software engineering) have provided encouragement, support, and guidance in developing quality curricula that are viable and dynamic. Degree programs have moved from language- and coding-centered curricula to those that emphasize theory, abstraction, and design. To address the problems in software development the ACM/IEEE-CS Joint Task Force on Computing Curricula has produced guidelines for curricula in computer engineering, computer science, information systems, information technology, and software engineering. The software engineering guidelines, SE2004 ( *Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering* ) [7], provide information and guidance on various curriculum issues: objectives, content, organization, courses, and pedagogy.

More recently the ILERT (International Learning Environment for Real-Time Software Intensive Control System) project [8] has been involved in the creation of an international curriculum framework centered on RSIC (Real-Time Software-Intensive Control) systems. The ILERT study explores a mechanism for involving students from multilingual, geographically separated institutions in a coordinated educational experience. The ultimate objective is the creation of a RSIC curriculum model, which can be used by engineering schools both in the USA and the EU.

The purpose of this paper is to discuss how the work of the ILERT project might be used to develop an SE2004 software engineering curriculum which satisfies the RSIC framework. In the following sections we provide additional detail about ILERT, the SE2004 guide, and the RSIC curriculum framework. Then, this material is integrated to outline a curriculum which satisfies both the SE2004 and the RSIC frameworks.

## II. The Ilert Project

The analysis, design, implementation, administration, and assessment of international curricula will become increasingly important in the global community of the 21st century. In support of this critical issue, the European Commission and the US Department of Education have funded the ATLANTIS initiative to promote collaboration in higher education between European and American universities. One American (Embry-Riddle Aeronautical University, Daytona Beach, FL) and three European Universities (AGH Univer-

978-83-60810-14-9/08/$25.00 © 2008 IEEE

sity of Science and Technology, Krakow, Poland; Brno University of Technology, Czech Republic; and The University of Grenoble, France) are currently working on the framework of a new common curriculum in real time-software systems. This two-year project "Toward International Learning Environment for Real-Time Software Intensive Control Systems" (EC grant: 2006-4563/006 001, US grant: P116J060005, http://www.ilert.agh.edu.pl) was launched in January 2007. Project work is concerned with program objectives and outcomes, curriculum content and pedagogy, program administration (academic credit, course schedules, exchange of students and staff, etc.), and program assessment and accreditation. Thus far, the project has produced the following deliverables:

- An analysis of industry requirements for graduates in the RSIC domain
- An identification of the RSIC learning objectives and student outcomes
- An analysis of European Credit Transfer System (ECTS) and the mechanism of credit transfer at U.S. Colleges and Universities
- An identification of activities and data for program assessment and evaluation, and those issues and elements required to consider program accreditation
- A description of an international, interdisciplinary RSIC curriculum framework
- A preliminary design for a selected unit supporting the proposed RSIC curriculum.

For the analysis of industry requirements, a survey of USA and European industry engaged in real-time software-intensive control systems was conducted [9]. The survey consisted of two parts: General Skills and Attitudes (10 items), and Technical Knowledge Areas (15 items). For example, one of the Technical Items was "Knowledge of software design and development concepts, methods and tools". Respondents were ask to select "Essential, Important, Unrelated, or Unimportant", with a possibility to provide comment.

Survey results from 43 companies were analyzed and summarized. In the General Skills part, the highest rated skills as follows:

- Work as a part of a multidisciplinary team
- Analyze, understand and define the problem

For the Technical Knowledge part, the following were rated the highest:

- Software design and development concepts, methods and tools
- System specification and design methods

Project work continues on experimental concurrent delivery of the designed RSIC unit at the four partner sites. Finally, the project will provide reflection on a process and methodology for creation of multidisciplinary, transatlantic engineering programs, including guidelines for extension of the approach to other engineering disciplines.

## III. The RSIC Curriculum Framework

This section provides information on the organization and content of the RSIC curriculum framework. The framework is a high-level curriculum specification that is detailed enough to guide the development of a RSIC program, which supports the RSIC objectives and outcomes, and yet is flexible enough to account for specializations, constraints, and requirements of various programs, institutions, and regions.

Table I.
RSIC Components

| **Software Engineering (SoftEng-)** |
|---|
| Software engineering concepts and practices, software lifecycle models, project management, software processes, software construction methods and practices, software modeling and formal representation; software requirements; software architectural and module design; testing and quality assurance; software maintenance; and notations and tools. |
| **Digital Systems (DigSys-)** |
| Digital system concepts/operation, design of combinatorial/sequential circuits, concepts and operation of microcontrollers/microprocessors, assembly language, rudimentary interfacing and exception handling, large scale integration devices and tools, interfacing, advanced memory management, fault tolerant hardware. |
| **Computer Control (CompCtrl)** |
| Concepts of feedback control, time and frequency domains, continuous and discrete models of dynamical systems, state analysis, stability, controllability and observability, controller design, implementing control algorithms in real-time, integrated control design and implementation use of analysis and design tools. |
| **Real-Time Systems (RTSys)** |
| Timing and dependability properties of software intensive systems, RTOS concepts and applications, concurrency, synchronization and communication, scheduling, reliability and safety, etc. |
| **Networking (Network)** |
| Data communication, network topology, analysis and design, information security, algorithms, encryption, bus architectures, wireless, etc. distributed control and monitoring |
| **System Engineering (SysEng)** |
| System engineering concepts, principles, and practices; system engineering processes (technical and management); system requirements, system design, system integration, and system testing; special emphasis on the development of a RSIC system and the integration of RSIC system elements. |

The basic organizational unit for the framework is a RSIC "component". A RSIC component is a curriculum unit which covers theory, knowledge and practice which supports the RSIC curriculum objective and outcomes. Table I describes the RSIC components in six identified RSIC areas: Software Engineering, Digital Systems, Computer Control, Real-Time Systems, Networking, and Systems Engineering.

The RSIC Curriculum Framework does not specify the way in which component topics might be formed into modules or courses. Component topics might be focused in one or two courses, or spread among several courses, along with other non-RSIC topics. Depending on the course rigor and the required prerequisite knowledge, the material can be at either a basic or an advanced level. The curriculum framework includes more detailed specifications of each component: prerequisite knowledge, component learning objectives, information about required facilities and equipment, and guidelines and suggestions for course design and delivery. Table II is an example of one such component specification. The full details of each competent specification will be included in the final ILLERT project report.

The RSIC curriculum framework also makes recommendations about non-RSIC courses or units that should be part of a RSIC program, as prerequisite courses or to supplement the components as part of a full degree program. The recommendations call for courses in the following areas:

- Mathematics (Differential and Integral Calculus, Differential Equations, Discrete Mathematics, Statistics, Linear Algebra)
- Physics (mechanics, E&M, thermo, fluids)
- Electrical Engineering (circuit analysis, basic electronics)

Table II.
Software Engineering

| | |
|---|---|
| **Description** | Software engineering concepts and practices, software lifecycle models, project management, software processes, software construction methods and practices. |
| **Prerequisite Knowledge** | Ability to design, implement and test small programs (100 lines of code), written in a commonly used high-level programming language. |
| **Learning Outcomes** | Upon completion of this component, students should be able to<br>• Describe the major problems in the development of a large, complex software system.<br>• Describe and discuss issues, principles, methods and technology associated with software engineering theory and practices (e.g., planning, requirements analysis, design, coding, testing, quality assurance, risk assessment, and configuration management).<br>• Working as part of a team, use a defined software development process to develop a high-quality modest sized software product (1000 lines of code).<br>• Describe issues, principles, methods and technology associated with the use of formal modeling in software engineering.<br>• Describe, discuss, and apply the commonly accepted principles of software quality assurance (reviews, inspections and testing).<br>• Apply requirements engineering principles of elicitation, analysis, and modeling to the development of a requirements specification.<br>• Describe and analyze different software architectures views and styles.<br>• Describe and discuss the structured and object-oriented design methodologies.<br>• Describe and discuss the principles, methods and practices of software evolution.<br>• Show capability with various software engineering tools used for formal software modeling, requirements engineering and software design. |
| **Facilities and Equipment** | No special equipment or laboratory is required for this component.<br>Students will need access to a computer system equipped with a program development environment (such as Eclipse).<br>There may be a need for process, management and formal modeling tools; but, typically word processors, spreadsheets and simple scheduling tools should be sufficient. |
| **Guidelines and Suggestions** | The emphasis in this component is for students to know and understand how large complex systems should be developed, not, at this point, to be able to develop such systems. For instance, they should understand and be able to describe and discuss the activities and practices that take place in each software development life-cycle phase; they should understand the importance of requirements and the problems that ensue if requirements are not properly elicited and specified; they should understand the various elements of project management; they should come to realize that testing is not the only way to ensure quality; they should come to see software development as an engineering discipline; and they should understand the importance of discipline and process to the development of software.<br>Case studies are particularly helpful in teaching software engineering principles – when students study examples of actual requirements, design, test and planning documents, they better understanding the nature of software engineering and what it takes to develop high-quality software products. There are many introductory software engineering textbooks that contain such examples (Pressman, Somerville, Lethbridge, Pfleeger)<br>This component should include a software team project. It is extremely important that the software product developed be modest in scope and functionality. The purpose of the project is for students to learn about working as part of the team, to experience the software-life cycle, to see how to assure quality, to use planning and process procedures, to document the team's work, and to appreciate the difficulty of developing a high-quality software product. The emphasis should be on teamwork, quality and process.<br>This component is intended to provide more in-depth coverage of software engineering topics than the Basic Level component. However, the coverage still must be at a level that can be covered in one or two courses. It is not intended that there would be an in-depth study of each of the separate areas of formal modeling, requirements, architecture and design, and quality and testing. Such in-depth coverage would require three or four course of study.<br>A development project could involve the creation of a requirements and architecture specification for a software system with more complexity than the Basic level team project.<br>This component would benefit from the use of case study documents (requirements, architecture, module design, code, test plans, project plans, etc.) for a reasonably complex system. Analysis and maintenance problems focused on the case study documents would be helpful in achieving the component learning outcomes. |

TABLE III.
SEEK KNOWLEDGE AREAS

| Knowledge Area | Hours |
|---|---|
| Computing Essentials | 172 |
| Mathematical & Engineering Fundamentals | 89 |
| Professional Practice | 35 |
| Software Modeling & Analysis | 53 |
| Software Design | 45 |
| Software V & V | 42 |
| Software Evolution | 10 |
| Software Process | 13 |
| Software Quality | 16 |
| Software Management | 19 |
| total hours | 494 |

- Engineering Economics
- Introduction to Computer Science with Programming

## IV. THE SE 2004 CURRICULUM FRAMEWORK

The software engineering guidelines document, SE2004 [7], provides a comprehensive and detailed set of material to support the development of an undergraduate curriculum in software engineering. Specifically it includes chapters on the following:

- The Software Engineering Discipline
- Guiding Principles
- Overview of Software Engineering Education Knowledge (SEEK)
- Guidelines for SE Curriculum Design and Delivery
- Courses and Course Sequences
- Adaptation to Alternative Environments
- Program Implementation and Assessment

The SEEK describes the body of knowledge that is appropriate for an undergraduate program in software engineering. It designates “core” material which SE2004 recommends is necessary for anyone to obtain an undergraduate degree in the field. It designates Bloom’s levels for the knowledge units [10] and makes a time allocation of “contact” hours. Table III lists the knowledge areas that make up the SEEK and indicates the minimum total time recommended for each area. The SE2004 contains a more detailed description of the content and organization of each area.

In addition to the core materials, undergraduates are encouraged to specialize in some area related to software engineering application. The following specialties are presented in the SE 2004 volume:

a) Network-centric systems
b) Information systems and data processing
c) Financial and e-commerce systems
d) Scientific systems
e) Telecommunications systems
f) Fault tolerant and survivable systems
g) Highly secure systems
h) Safety critical systems
i) Embedded and real-time systems
j) Biomedical systems
k) Avionics and vehicular systems
l) Industrial process control systems

TABLE IV.
SE2004 CORE COURSES

| Number | Title |
|---|---|
| SE101 | Software Engineering and Computing I |
| SE102 | Software Engineering and Computing II |
| SE103 | Software Engineering and Computing III |
| SE211 | Software Construction |
| SE212 | Software Engineering Approach to Human Computer Interaction |
| SE311 | Software Design and Architecture |
| SE321 | Software Quality Assurance and Testing |
| SE322 | Software Requirements Analysis |
| SE323 | Software Project Management |
| SE400 | Software Engineering Capstone Project |

Of these, several certainly correlate strongly with the RCIS components. This provides the motivation and rationale for proposing a framework for an SE 2004 RSIC curriculum.SE2004 also includes a set of courses and their descriptions which, when grouped together, provide for a set of courses which cover the SEEK core knowledge. Table IV describes one such set of courses.

Courses SE101, SE102 and SE103 provide an overview of software engineering with topics typically included in introductory courses in computer science, low level design, and programming. Each course, except for SE400, was envisioned to be offered over approximately 14 weeks, with 3 contact hours per week. SE400 covers a full academic year of work. Other schedules and timelines are possible. SE2004 provides detailed descriptions of each course.

In addition, SE2004 contains several models (or patterns) of how the courses could be arranged into a full three or four year curriculum. The models include three-year and four-year patterns, with versions for North America, Europe, Japan, Australia, and Israel, Table V depicts a four year “North American” curriculum pattern. In addition to the SE core courses in Table V (shaded gray), the SE2004 also contains course descriptions for supporting courses such as Discrete Math I and II, Data Structures and Algorithms, Database Systems, Computer Architecture, Operating Systems and Networking, etc.

## V. A COMBINED SE 2004-RSIC CURRICULUM

In this section we present a curriculum that incorporates the RSIC requirements within the SE2004 requirements. Table VI outlines a four year RSIC-SE2004 curriculum. It represents a modification of Table V: all of the software engineering core courses were retained; and technical electives and science electives items were replaced with courses which

TABLE V
SE2004 NORTH AMERICAN CURRICULUM PATTERN

| Year 1 | | Year 2 | | Year 3 | | Year 4 | |
|---|---|---|---|---|---|---|---|
| **Sem 1A** | **Sem 1B** | **Sem 2A** | **Sem 2B** | **Sem 3A** | **Sem 3B** | **Sem 4A** | **Sem 4B** |
| SE101 | SE102 | SE 200 | SE 211 | SE 212 | SE 311 | SE400 | SE400 |
| Dis Math I | Dis Math II | Data Str & Alg | Database | SE 321 | SE 322 | SE 323 | Tech Elect |
| Calc 1 | Calc 2 | Physics 1 | Physics 2 | Comp Arch | Prof SE Practice | OS & Network | Tech Elect |
| Gen Ed | Gen Ed | Gen Ed | Statistics | Sci Elect | Tech Elect | Eng Econ | Open Elect |
| Gen Ed | Gen Ed | Gen Ed | Sci Elect | Sci Elect | Gen Ed | Gen Ed | Open Elect |

cover the RSIC curriculum components.

The courses in bold type in Table VI represent courses that directly address the requirements of the RSIC components: for example, **DigSys-1** and **DigSys-2** would cover the Digital Systems Component. Notice that **SoftEng** is designated in several courses. This was necessary in order to cover all the RSIC SE advanced topics. Due to the nature of this specific SE program, such topics would be covered in more depth than strictly required by the RSIC Framework. In addition, a number of course were added in order to support prerequisite requirements for RSIC courses: differential equations, electrical engineering and linear algebra.

The course “Prof **RSIC** Practice” is a replacement for the course NT291 (Professional Software Engineering Practice - knowledge, skills, and attitudes that software engineers must possess to practice software engineering in a professional, responsible, and ethical manner.). The RSIC course would contain much of the material from NT291; however, it would also furnish students with material and activities that support two of the non-technical RSIC curriculum outcomes:

- An ability to work effectively in an international environment
- An understanding of the impact of engineering solutions in a global and societal context

One other important item in Table VI is that **SysEng** was designated as part of the SE400. SE400 is a capstone project course and it is envisioned that the project will be a real-time software intensive control system. It relies on and will bring together the knowledge and practices learned by students in the other RSIC courses. Although students will have been introduced to some system concepts in other courses, SE400 is the ideal place to focus on system issues: requirements require system level decisions about allocation and specification; the system architecture will involve software and hardware subsystems and components; systems quality assurance measures will have to be instituted; and the procedures and activities’ involved in system integration will be critical.

Although Table VI provides an outline of a RSIC-SE400 curriculum, much more analysis and detail is needed to support implementation of such a curriculum. However, it is the hope of the authors that this paper, along with the other work of the ILERT project, will motivate and support faculty who wish to create and implement educational programs which will improve the development of RSIC systems.

## VI.CONCLUSION

Creation of RSIC systems engages a large variety of engineering disciplines. Due to worldwide implementation of such systems, a well prepared workforce of scientists and engineers is required. They must be able to work cooperatively in multi-disciplinary and international settings. The software intensive nature of RSIC systems require engineers who understand and can use the software engineering knowledge and practices required to build such systems. The authors feel that the RSIC-SE2004 the potential to have broad impact on the future of engineering education and on the efficient and effective development of RSIC systems.

In addition, the process and format of the RSIC-SE2004

TABLE VI
A RSIC-SE2004 CURRICULUM

| Year 1 | | Year 2 | | Year 3 | | Year 4 | |
|---|---|---|---|---|---|---|---|
| **Sem 1A** | **Sem 1B** | **Sem 2A** | **Sem 2B** | **Sem 3A** | **Sem 3B** | **Sem 4A** | **Sem 4B** |
| SE101 | SE102 | SE 200<br>**SoftEng-1** | SE 211<br>**SoftEng-2** | SE 212<br>**SoftEng-3** | SE 311<br>**SoftEng-5** | SE400<br>**SysEng-1** | SE400<br>**SysEng-2** |
| Dis Math I | Dis Math II | Data Str & Alg | **DigSys-1** | SE 321<br>**SoftEng-4** | SE 322<br>**SoftEng-6** | SE 323<br>**SoftEng-7** | Tech Elect |
| Calc 1 | Calc 2 | Physics 1 | Physics 2 | **DigSys-2** | **RTSys** | Prof **RSIC** Practice | Tech Elect |
| Gen Ed | Gen Ed | Gen Ed | Statistics | **CompCtrl** | **Network** | Eng Econ | Open Elect |
| Gen Ed | Gen Ed | Diff Eqns | Elec Eng | OS & Network | Lin Alg | Gen Ed | Gen Ed |

curriculum framework could be used as a model for the development of other “integrated” curricula (e.g., a RSIC curriculum for computer engineering, or a RSIC curriculum for control engineering).

## Acknowledgment

The authors would like to thank the European Commission, the US Department of Education and the ATLANTIS program for their support of ILERT project: "Toward International Learning Environment for Real-Time Software Intensive Control Systems" (EC grant: 2006-4563/006 001, US grant: P116J060005, http://www.ilert.agh.edu.pl).

## References

[1] M. Cusumano, A. MacCormack, C. Kemerer, and B. Crandall, “Software Development Worldwide: The State of the Practice”, *IEEE Software*, pp.28-34, vol. 20, no. 6, Nov./Dec. 2003.

[2] N. Davis, N. and J. Mullaney, *The Team Software Process (TSP) in Practice: A Summary of Recent Results*, CMU/SEI-2003-TR-014, Software Engineering Institute, Carnegie Mellon University, Sep. 2003.

[3] C. Jones, “Variations in Software Development Practices”, *IEEE Software*, pp.22-37, vol. 20, no. 6, Nov./Dec. 2003.

[4] W. Humphrey and T. Hilburn, “The Impending Changes in Software Education”, *IEEE Software*, Vol. 19 , No. 5, pp. 22-24, Sep. / Oct., 22 – 24, 2002

[5] J. Knight, N. Leveson, “Software and Higher Education”, Inside Risks Column, *Communications of the ACM*, p. 160, vol. 49, no. 1, Jan. 2006.

[6] L. Long, “The Critical Need for Software Engineering Education”, *CrossTalk: The Journal of Defense Software Engineering*, pp. 6-10, Jan. 2006. (http://www.stsc.hill.af.mil/crosstalk/2008/01/0801-Long.pdf)

[7] ACM/IEEE-CS Joint Task Force on Computing Curricula, *Software Engineering 2004,Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, Aug. 2004. (http://www.acm.org/education/curricula.html)

[8] W. Grega, A. Kornecki, M. Sveda, and J. Thiriet, “Developing Interdisciplinary and Multinational Software Engineering Curriculum”, *Proceedings of the ICEE’07*, Coimbra, Portugal, Sep. 3-7, 2007.

[9] A. Pilat, A. Kornecki, J. Thiriet, W. Grega, and M. Sveda, “Industry Feedback on Skills and Knowledge in Real-Time Software Engineering”, Proceedings of 19th EAEEIE Annual Conference, Tallinn, Estonia, Jun29 - Jul 2, 2008.

[10] B. S. Bloom, Editor, *Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain*, Longmans, 1956.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# You Can’t Get There From Here! Problems and Potential Solutions in Developing New Classes of Complex Computer Systems

Michael G. Hinchey
Lero—the Irish Software
Engineering Research Centre
University of Limerick
Limerick, Ireland
mike.hinchey@lero.ie

James L. Rash, Walter F. Truszkowski
NASA Goddard Space Flight Center
Systems Engineering Division
Greenbelt, MD 21035 USA
{james.l.rash, walter.f.truszkowski}@nasa.gov

Roy Sterritt
School of Computing and Mathematics
University of Ulster
Northern Ireland
r.sterritt@ulster.ac.uk

Christopher A. Rouff
Lockheed Martin Advanced
Technology Laboratories
Arlington, VA 22203 USA
christopher.rouff@lmco.com

***Abstract*—The explosion of capabilities and new products within the sphere of Information Technology (IT) has fostered widespread, overly optimistic opinions regarding the industry, based on common but unjustified assumptions of quality and correctness of software. These assumptions are encouraged by software producers and vendors, who at this late date have not succeeded in finding a way to overcome the lack of an automated, mathematically sound way to develop correct systems from requirements. NASA faces this dilemma as it envisages advanced mission concepts that involve large swarms of small spacecraft that will engage cooperatively to achieve science goals. Such missions entail levels of complexity that beg for new methods for system development far beyond today’s methods, which are inadequate for ensuring correct behavior of large numbers of interacting intelligent mission elements. New system development techniques recently devised through NASA-led research will offer some innovative approaches to achieving correctness in complex system development, including autonomous swarm missions that exhibit emergent behavior, as well as general software products created by the computing industry.**

## I. Introduction

SOFTWARE has become pervasive. We encounter it in our everyday lives: the average electric razor contains the equivalent of more than 100,000 lines of code, several high-end cars contain more software than the onboard systems of the Space Shuttle. We are reliant on software for our transportation and entertainment, to wash our clothes and cook our meals, and to keep us in touch with the outside world via the Internet and our mobile phones.

The Information Technology industry, driven by software development, has made remarkable advances. In just over half a century, it has developed into a trillion-dollar-per-year industry, continually breaking its own records [17], [27].

Some breathtaking statistics have been reported for the hardware and software industries [16], [46]:

- The Price-to-Performance ratio halves every 18 months, with a 100-fold increase in performance every decade.
- Performance progress in the next 18 months will equal *all* progress made to date.
- New storage available equals the sum of all previously available storage *ever*.
- New processing capability equals the sum of all previous processing power.

Simultaneously, a number of flawed assumptions have arisen regarding the way we build both software and hardware systems [38], [46], which include:

- Human beings can achieve perfection; they can avoid making mistakes during installation, maintenance and upgrades.
- Software will eventually be bug-free; the focus of companies has been to hire better programmers, and the focus of universities is to better train software engineers in development lifecycle models.
- Mean-time between failure (MTBF) is already very large (approximately 100 years) and will continue to increase.
- Maintenance costs are a function of the purchase price

This paper was previously published in Proc. Eighth International Conference on Integrated Design and Process Technology (IDPT), 2005. Reprinted with permission.

This work is funded in part by Science Foundation Ireland grant 03/CE2/I303_1 to Lero—the Irish Software Engineering Research Centre (www.lero.ie); by the NASA Office of Safety and Mission Assurance, under its Software Assurance Research Program project *Formal Approaches to Swarm Technologies* (FAST), administered by the NASA IV&V Facility; by the Office of Technology Transfer, NASA Goddard Space Flight Center; by the NASA Software Engineering Laboratory, NASA Goddard Space Flight Center; and by the University of Ulster Computer Science Research Institute and the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme under the European Union Peace II initiative.

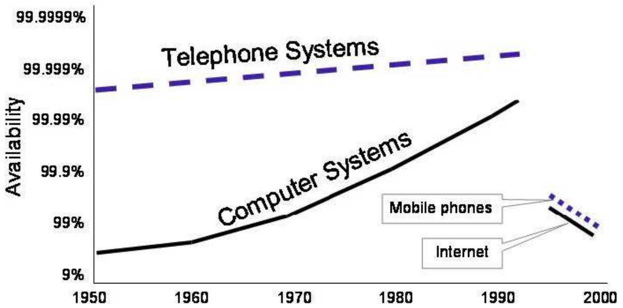


Fig. 1. Contrasting availability of Telephone Systems, Computer Systems, Internet, and Mobile Phones.

of hardware; and, as such, decreasing hardware costs (price/performance) results in decreases in maintenance costs.

## II. Software Problems

With the situation stated this way, many flawed assumptions regarding the IT industry come into view. The situation is even worse if we focus primarily on software. The Computing industry has failed to avoid software-related catastrophes. Notable examples include:

- Therac-25, where cancer patients were given lethal doses of radiation during radiation therapy [33].
- Ariane 5, where it was assumed that the same launch software used in the prior version (Ariane 4) could be reused. The result was the loss of the rocket within seconds of launch [34].
- The Mars Polar Lander, where failure to initialize a variable resulted in the craft crash landing on the Martian surface, instead of reverse thrusting and landing softly [29].

Progress in software regularly lags behind hardware. In the last decade, for example, two highly software-intensive applications, namely Internet communications and mobile phone technology, have suffered reduced availability and increased *down time*, while their hardware counterparts, computer hardware and telephony systems, have continued to improve. Figure 1 illustrates this trend [17].

### A. An Historic Problem

The realization that software development has lagged greatly behind hardware is hardly a new one [6], nor is the realization that our software development processes have some severe deficiencies.

Brooks, in a widely quoted and much-referenced article [7], warns of complacency in software development. He stresses that, unlike hardware development, we cannot expect to achieve great advances in productivity in software development unless we concentrate on more appropriate development methods. He highlights how software systems can suddenly turn from being well-behaved to behaving erratically and uncontrollably, with unanticipated delays and increased costs. Brooks sees software systems as “werewolves” and rightly points out that there is no single technique, no Silver Bullet, capable of slaying such monsters [6].

On the contrary, more and more complex systems are run on highly distributed, heterogeneous networks, subject to strict performance, fault tolerance, and security constraints, all of which may conflict. Many engineering disciplines must contribute to the development of complex systems in an attempt to satisfy all of these requirements. No single technique is adequate to address all issues of complex system development; rather, different techniques must be applied at different stages of development (and throughout the development process) to ensure unambiguous requirements statements, precise specifications that are amenable to analysis and evaluation, implementations that satisfy the requirements and various (often conflicting) goals, re-use, re-engineering and reverse engineering of legacy code, appropriate integration with existing systems, ease-of-use, predictability, dependability, maintainability, fault tolerance, etc. [6].

Brooks [7] differentiates between the *essence* (that is, problems that are necessarily inherent in the nature of software) and *accidents* (that is, problems that are secondary and caused by current development environments and techniques). He points out the great need for appropriate means of coming to grips with the conceptual difficulties of software development—that is, for appropriate emphasis on specification and design, rather than on coding and testing.

In his article [7], he highlights some successes that have been achieved in gaining improvements in productivity, but points out that these address problems in the current development process, rather than the problems inherent in software itself. In this category, he includes: the advent of high-level programming languages, time-sharing, and unified programming environments. Object-oriented programming, techniques from artificial intelligence, expert systems, automatic programming, program verification, and the advent of workstations, he sees as non-bullets, as they will not help in slaying the werewolf.

He sees software reuse, rapid prototyping, incremental development, and the employment of top-class designers as potential starting points for the Silver Bullet, but warns that none in itself is sufficient.

Brooks’ article has been very influential, and remains one of the classics of software engineering. His viewpoint has been criticized, however, as being overly pessimistic and for failing to acknowledge some promising developments [6].

Harel, in an equally influential paper, written as a rebuttal to Brooks [19], points to developments in Computer-Aided Software Engineering (CASE) and visual formalisms [18] as potential *bullets*. Harel’s view is far more optimistic. He writes five years after Brooks, and has seen the developments in that period. The last forty years of system development have been equally difficult, according to Harel, and, using a conceptual vanilla framework, the development community has devised means of overcoming many difficulties. As we address more complex systems, Harel argues that we must devise similar frameworks that are applicable to the classes of system we are developing.

Harel, along with many others, including the authors of this paper, believes that appropriate techniques for modeling must have a rigorous mathematical semantics, and appropriate means for representing constructs. This differs greatly from Brooks, who sees representational issues as mainly *accidental*.

## III. New Challenges for Software Engineering

Clearly there have been significant advances in software engineering tools, techniques, and methods, since the time of Brooks' and Harel's papers. In many cases, however, the advantages of these developments have been mitigated by corresponding increases in demand for greater, more complex functionality, stricter constraints on performance and reaction times, and attempts to increase productivity and reduce costs, while simultaneously pushing systems requirements to their limits. NASA, for example, continues to build more and more complex systems, with impressive functionality, and increasingly autonomous behavior. In the main, this is essential. NASA missions are pursuing scientific discovery in ways that require autonomous systems. While manned exploration missions are clearly in NASA's future (such as the Exploration Initiative's plans to return to the moon and put Man on Mars), several current and future NASA missions, for reasons that we will explain below, necessitate autonomous behavior by unmanned spacecraft.

We will describe some of the challenges for software engineering emerging from new classes of complex systems being developed by NASA and others. We will discuss these in Section III-A with reference to a NASA concept mission that is exemplary of many of these new systems. Then, in Section IV we will present some techniques that we are addressing, which may lead towards a Silver Bullet.

### A. *Challenges of Future NASA Missions*

Future NASA missions will exploit new paradigms for space exploration, heavily focused on the (still) emerging technologies of autonomous and autonomic systems. Traditional missions, reliant on one large spacecraft, are being superceded or complemented by missions that involve several smaller spacecraft operating in collaboration, analogous to swarms in nature. This offers several advantages: the ability to send spacecraft to explore regions of space where traditional craft simply would be impractical, increased spatial distribution of observations, greater redundancy, and, consequently, greater protection of assets, and reduced costs and risk, to name but a few. Planned missions entail the use of several unmanned autonomous vehicles (UAVs) flying approximately one meter above the surface of Mars, covering as much of the surface of Mars in seconds as the now famous Mars rovers did in their entire time on the planet; the use of armies of tetrahedral walkers to explore the Mars and Lunar surface; constellations of satellites flying in formation; and the use of miniaturized pico-class spacecraft to explore the asteroid belt.

These new approaches to exploration missions simultaneously pose many challenges. The missions will be unmanned and necessarily highly autonomous. They will also exhibit all of the classic properties of autonomic systems, being self-protecting, self-healing, self-configuring, and self-optimizing. Many of these missions will be sent to parts of the solar system where manned missions are simply not possible, and to where the round-trip delay for communications to spacecraft exceeds 40 minutes, meaning that the decisions on responses to problems and undesirable situations must be made *in situ* rather than from ground control on Earth.

Verification and Validation (V&V) for complex systems still poses a largely unmet challenge in the field of Computing, yet the challenge is magnified with increasing degrees of system autonomy. It is an even greater open question as to the extent to which V&V is feasible when the system possesses the ability to adapt and learn, particularly in environments that are dynamic and not specially constrained. Reliance on testing as the primary approach to V&V becomes untenable as systems move towards higher levels of complexity, autonomy, and adaptability in such environments. Swarm missions will fall into this category, and an early concern in the design and development of swarms will be the problem of predicting, or at least bounding, and controlling emergent behavior.

The result is that formal specification techniques and formal verification will play vital roles in the future development of NASA space exploration missions. The role of formal methods will be in the specification and analysis of forthcoming missions, enabling software assurance and proof of correctness of the behavior of these systems, whether or not this behavior is emergent (as a result of composing a number of interacting entities, producing behavior that was not foreseen). Formally derived models may also be used as the basis for automating the generation of much of the code for the mission. To address the challenge in verifying the above missions, a NASA project, Formal Approaches to Swarm Technology (FAST), is investigating the requirements of appropriate formal methods for use in such missions, and is beginning to apply these techniques to specifying and verifying parts of a future NASA swarm-based mission.

### B. *ANTS: A NASA Concept Mission*

The Autonomous Nano-Technology Swarm (ANTS) mission will involve the launch of a swarm of autonomous pico-class (approximately 1kg) spacecraft that will explore the asteroid belt for asteroids with certain characteristics. Figure 2 gives an overview of the ANTS mission [47]. In this mission, a transport ship, launched from Earth, will travel to a point in space where gravitational forces on small objects (such as pico-class spacecraft) are all but negligible. Objects that remain near such a point (termed a Lagrangian point) are in a stable orbit about the Sun and will have a fixed geometrical relationship to the Sun-Earth system. From the transport ship positioned at such a point, 1000 spacecraft that have been assembled en route from Earth will be launched into the asteroid belt.

Because of their small size, each ANTS spacecraft will carry just one specialized instrument for collecting a specific type of data from asteroids in the belt. As a result, spacecraft

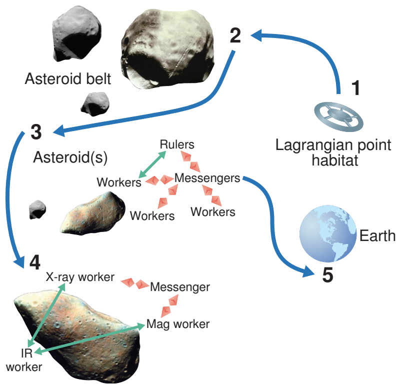


Fig. 2. NASA's Autonomous Nano Technology Swarm (ANTS) mission scenario.

must cooperate and coordinate using a hierarchical social behavior analogous to colonies or swarms of insects, with some spacecraft directing others. To implement this mission, a heuristic approach is being considered that provides for a social structure to the swarm based on the above hierarchy. Artificial intelligence technologies such as genetic algorithms, neural nets, fuzzy logic and on-board planners are being investigated to assist the mission to maintain a high level of autonomy. Crucial to the mission will be the ability to modify its operations autonomously to reflect the changing nature of the mission and the distance and low-bandwidth communications back to Earth.

Approximately 80 percent of the spacecraft will be workers that will carry the specialized instruments (e.g., a magnetometer, x-ray, gamma-ray, visible/IR, neutral mass spectrometer) and will obtain specific types of data. Some will be coordinators (called rulers) that have rules that decide the types of asteroids and data the mission is interested in, and that will coordinate the efforts of the workers. The third type of spacecraft are messengers that will coordinate communication between the rulers and workers, and communications with the Earth ground station, including requests for replacement spacecraft with specialized instruments as these are required. The swarm will form sub-swarms under the direction of a ruler, which contains models of the types of science that it wants to perform. The ruler will coordinate workers each of which uses its individual instrument to collect data on specific asteroids and feed this information back to the ruler who will determine which asteroids are worth examining further. If the data matches the profile of a type of asteroid that is of interest, an imaging spacecraft will be sent to the asteroid to ascertain the exact location and to create a rough model to be used by other spacecraft for maneuvering around the asteroid. Other teams of spacecraft will then coordinate to finish the mapping of the asteroid to form a complete model.

### C. Problematic Issues

*1) Size and Complexity:* While the use of a swarm of miniature spacecraft is essential for the success of ANTS, it simultaneously poses several problems in terms of adding significantly to the complexity of the mission.

The mission will launch 1000 pico-class spacecraft, many of which possibly will be destroyed by collisions with asteroids, since the craft, having no means of maneuvering other than solar sails, will be very limited in their collision-avoidance capabilities. The several hundred surviving spacecraft must be organized into effective groups that will collect science data and make decisions as to which asteroids warrant further investigation. These surviving spacecraft effectively form a wireless sensor network [23] tens of millions of miles from Earth. The overhead for communications is clearly significant.

To keep the spacecraft small, each craft only carries a single instrument. That is why several craft must coordinate to investigate particular asteroids and collect different types of science data. Again, while miniaturization is important, the use of such a scheme has a major drawback: we have no *a priori* knowledge as to which instruments will be lost during normal operations (where we expect to regularly lose craft due to collisions).

The need to identify lost capabilities and instruments, and then replace them, presents an extremely complex problem. In the case of lost messengers and rulers, other craft may be *promoted* to replace them. It is merely the software that differentiates messengers and rulers from other workers, so mobile code serves to overcome this problem. When an instrument is lost, however, we have a rather different problem. A worker with a damaged instrument can be reserved for use as a ruler, and another spacecraft with an identical instrument can replace it.

An alternative would be add more features (instruments) into each spacecraft, but this would increase both their size (a problem in such a constrained environment) and their power requirements. The addition of features, of course, also increases complexity, as identified by Lawson [32].

*2) Emergent Behavior:* In swarm-based systems, interacting agents (often homogeneous or near homogeneous agents) are developed to take advantage of their emergent behavior. Each of the agents is given certain parameters that it tries to maximize. Bonabeau et al. [4], who studied self-organization in social insects, state that "complex collective behaviors may emerge from interactions among individuals that exhibit simple behaviors" and describe emergent behavior as "a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components."

Intelligent swarms [3] use swarms of simple intelligent agents. Swarms have no central controller: they are self-organizing based on the emergent behaviors of the simple interactions. There is no external force directing their behavior and no one agent has a global view of the intended macroscopic behavior. Though current NASA swarm missions differ from true swarms as described above, they do have many of the same attributes and may exhibit emergent behavior. In addition, there are a number of US government projects that are looking at true swarms to accomplish complex missions.

*3) Autonomy:* Autonomous operation is essential for the success of the ANTS mission concept.

Round trip communications delays of up to 40 minutes, and limited bandwidth on communications with Earth, mean that effective control from the ground station is impossible. Ground controllers would not be able to react sufficiently quickly during encounters with asteroids to avoid collisions with asteroids and even other ANTS spacecraft. Moreover, the delay in sending instructions to the spacecraft would be so great that situations would likely have changed dramatically by the time the instructions were received.

But autonomy implies absence of centralized control. Individual ANTS spacecraft will operate autonomously as part of a subgroup under the direction of that subgroup's *ruler*. That ruler will itself autonomously make decisions regarding asteroids of interest, and formulate plans for continuing the mission of collecting science data. The success of the mission is predicated on the validity of the plans generated by the rulers, and requires that the rulers generate sensible plans that will collect valid science data, and then make valid informed decisions.

That autonomy is possible is not in doubt. What is in doubt is that autonomous systems can be relied upon to operate correctly, in particular in the absence of a full and complete specification of what is required of the system.

*4) Testing and Verification:* As can be seen from the brief exposition above, ANTS is a highly complex system that poses many significant challenges. Not least amongst these are the complex interactions between heterogeneous components, the need for continuous re-planning, re-configuration, and re-optimization, the need for autonomous operation without intervention from Earth, and the need for assurance of the correct operation of the mission.

As mission software becomes increasingly more complex, it also becomes more difficult to test and find errors. Race conditions in these systems can rarely be found by inputting sample data and checking whether the results are correct. These types of errors are time-based and only occur when processes send or receive data at particular times, or in a particular sequence, or after learning occurs. To find these errors, the software processes involved have to be executed in all possible combinations of states (state space) that the processes could collectively be in. Because the state space is exponential (and sometimes factorial) to the number of states, it becomes untestable with a relatively small number of processes. Traditionally, to get around the state explosion problem, testers have artificially reduced the number of states of the system and approximated the underlying software using models.

One of the most challenging aspects of using swarms is how to verify that the emergent behavior of such systems will be proper and that no undesirable behaviors will occur. In addition to emergent behavior in swarms, there are also a large number of concurrent interactions between the agents that make up the swarms. These interactions can also contain errors, such as race conditions, that are very difficult to ascertain until they occur. Once they do occur, it can also be very difficult to recreate the errors since they are usually data and time dependent.

As part of the FAST project, NASA is investigating the use of formal methods and formal techniques for verification and validation of these classes of mission, and is beginning to apply these techniques to specifying and verifying parts of the ANTS concept mission. The role of formal methods will be in the specification and analysis of forthcoming missions, while offering the ability to perform software assurance and proof of correctness of the behavior of the swarm, whether this behavior is emergent or not.

## IV. Some Potentially Useful Techniques

### A. Autonomicity

Autonomy may be considered as bestowing the properties of self-governance and self-direction, i.e., control over one's goals [15], [26], [43]. Autonomicity is having the ability to self-manage through properties such as self-configuring, self-healing, self-optimizing, and self-protecting. These are achieved through other self-properties such as self-awareness (including environment awareness), self-monitoring, and self-adjusting [45].

Increasingly, self-management is seen as the only viable way forward to cope with the ever increasing complexity of systems. From one perspective, self-management may be considered a specialism of self-governance, i.e., autonomy where the goals/tasks are specific to management roles [46]. Yet from the wider context, an autonomic element (AE), consisting of an autonomic manager and managed component, may still have its own specific goals, but also the additional responsibility of management tasks particular to the wider system environment.

It is envisaged that in an autonomic environment the AEs communicate to ensure a managed environment that is reliable and fault tolerant and meets high level specified policies (where a policy consists of a set of behavioral constraints or preferences that influences the decisions made by an autonomic manager [10]) with an overarching vision of system-wide policy-based self-management. This may result in AEs monitoring or *watching out for* other AEs. In terms of autonomy and the concern of undesirable emergent behavior, an environment that dynamically and continuously monitors can assist in detecting race conditions and reconfiguring to avoid damage (self-protecting, self-healing, self-configuring, etc.). As such, autonomicity becoming mainstream in the industry can only assist in improving techniques, tools, and processes for autonomy [44].

### B. Hybrid Formal Methods

The majority of formal notations currently available were developed in the 1970s and 1980s and reflect the types of distributed systems being developed at that time. Current distributed systems are evolving and may not be able to be specified in the same way that past systems have been developed. Because of this, it appears that many people are combining formal methods into integrated approaches to address some of the new features of distributed systems (e.g., mobile agents, swarms, and emergent behavior).

Integrated approaches have been very popular in specifying concurrent and agent-based systems. Integrated approaches often combine a process algebra or logic-based approach with a model-based approach. The process algebra or logic-based approach allows for easy specification of concurrent systems, while the model-based approach provides strength in specifying the algorithmic part of a system.

Some recent hybrid approaches include:

- CSP-OZ, a combination of CSP and Object-Z [11]
- Object-Z and Statecharts [8]
- Timed Communicating Object Z [13]
- Temporal B [5]
- Temporal Petri Nets (Temporal Logic and Petri Nets) [1]
- ZCCS, a combination of Z and CCS [14]

These and new hybrid formal methods are being investigated to address swarm and other complex NASA missions [41].

### *C. Automatic Programming*

For many years, automatic programming has referred, primarily, to the use of very high-level languages to describe solutions to problems, which could then be translated down and expressed as code in more traditional (lower level) programming languages. Parnas [36] implies that the term is glamorous, rather than having any real meaning, precisely because it is the solution that is being specified rather than the problem that must be solved. Brooks [7] supports this view, and equally criticizes the field of visual programming, arguing that it will never produce anything of value.

Writing just five years after Brooks, Harel [19] disagrees, faulting Brooks for failing to recognize advances in *visual formalisms*. Now, writing almost two decades after Brooks, we argue that automatic code generation is not only a viable option, it is essential to the development of the classes of complex system we are discussing here, and as exemplified by ANTS.

Autonomous and autonomic systems, exhibiting complex emergent behavior, cannot, in general, be fully specified at the outset. The roles and behaviors of the system will vary greatly over time. While we may try to write specifications that constrain the system, it is clear that not all behavior can be specified in advance. Consequently, the classes of system we are discussing will often require that code is generated, or modified, during execution. As a result, the classes of system we are describing here will *require* automatic code generation.

Several tools already exist that successfully generate code from a given model. Unfortunately, many of these tools have been shown to generate code, portions of which are never executed, or portions of which cannot be justified from either the requirements or the model. Moreover, existing tools do not and cannot overcome the fundamental inadequacy of all currently available automated development approaches, which is that they include no means to establish a provable equivalence between the requirements stated at the outset and either the model or the code they generate.

Traditional approaches to automatic code generation, including those embodied in commercial products such as Matlab [35], in system development toolsets such as the B-Toolkit [31] or the VDM++ toolkit [28], or in academic research projects, presuppose the existence of an explicit (formal) model of reality that can be used as the basis for subsequent code generation. While such an approach is reasonable, the advantages and disadvantages of the various modeling approaches used in computing are well known and certain models can serve well to highlight certain issues while suppressing other less relevant details [37]. It is clear that the converse is also true. Certain models of reality, while successfully detailing many of the issues of interest to developers, can fail to capture some important issues, or perhaps even the most important issues.

That is why, we believe, future approaches to automatic code generation must be based on Formal Requirements-Based Programming [39].

### *D. Formal Requirements Based Programming*

Requirements-Based Programming refers to the development of complex software (and other) systems, where each stage of the development is fully traceable back to the requirements given at the outset. In essence, Requirements-Based Programming takes Model-Based Development and adds a *front end* [40].

The difference is that Model-Based Development holds that emphasis should be placed on building a model of the system with such high quality that automatic code generation is viable. While this has worked well, and made automatic code generation feasible, there is still the large *analysis-specification* gap that remains unaddressed. Requirements-Based Programming addresses that issue and ensures that there is a direct mapping from requirements to design, and that this design (model) may then be used as the basis for automatic code generation.

There have been calls for the community to address Requirements-Based Programming, as it offers perhaps the most promising approach to achieving *correct* systems [20]. Although the use of Requirements-Based Programming does not specifically presuppose the existence of an underlying formalism, the realization that proof of correctness is not possible without formalism [2] certainly implies that Requirements-Based Programming should be formal.

In fact, Formal Requirements-Based Programming, coupled with a graphical representation for system requirements (e.g., UML use cases) possesses the features and advantages of a visual formalism described by Harel [18].

*1) R2D2C:* R2D2C, or Requirements-to-Design-to-Code [22], [39], is a NASA patent-pending approach to Requirements-Based Programming.

In R2D2C, engineers (or others) may write specifications as scenarios in constrained (domain-specific) natural language, or in a range of other notations (including UML use cases). These will be used to derive a formal model (Figure 3) that is

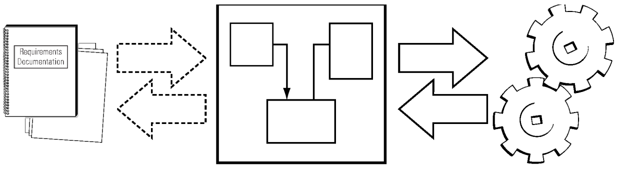


Fig. 3. The R2D2C approach, generating a formal model from requirements and producing code from the formal model, with automatic reverse engineering.

guaranteed to be equivalent to the requirements stated at the outset, and which will subsequently be used as a basis for code generation. The formal model can be expressed using a variety of formal methods. Currently we are using CSP, Hoare's language of Communicating Sequential Processes [24], [25], which is suitable for various types of analysis and investigation, and as the basis for fully formal implementations as well as for use in automated test case generation, etc.

R2D2C is unique in that it allows for full formal development from the outset, and maintains mathematical soundness through all phases of the development process, from requirements through to automatic code generation. The approach may also be used for reverse engineering, that is, in retrieving models and formal specifications from existing code, as shown in Figure 3. The approach can also be used to "paraphrase" (in natural language, etc.) formal descriptions of existing systems. In addition, the approach is not limited to generating high-level code. It may also be used to generate business processes and procedures, and we have been experimenting with using it to generate instructions for robotic devices that were to be used on the Hubble Robotic Servicing Mission (HRSM), which, at the time of writing, has not received a final go-ahead. We are also experimenting with using it as a basis for an expert system verification tool, and as a means of capturing domain knowledge for expert systems.

*2) R2D2C Technical Approach:* The R2D2C approach involves a number of phases, which are reflected in the system architecture described in Figure 4. The following describes each of these phases.

D1 Scenarios Capture: Engineers, end users, and others write scenarios describing intended system operation. The input scenarios may be represented in a constrained natural language using a syntax-directed editor, or may be represented in other textual or graphical forms.

D2 Traces Generation: Traces and sequences of atomic events are derived from the scenarios defined in phase D1.

D3 Model Inference: A formal model, or formal specification, expressed in CSP is inferred by an automatic theorem prover — in this case, ACL2 [30] — using the traces derived in phase D2. A deep[1] embedding of the laws of concurrency [21] in the theorem prover gives it sufficient knowledge of concurrency and of CSP to perform the inference. The embedding will be the topic of a future paper.

D4 Analysis: Based on the formal model, various analyses can be performed, using currently available commercial or public domain tools, and specialized tools that are planned for development. Because of the nature of CSP, the model may be analyzed at different levels of abstraction using a variety of possible implementation environments. This will be the subject of a future paper.

D5 Code Generation: The techniques of automatic code generation from a suitable model are reasonably well understood. The present modeling approach is suitable for the application of existing code generation techniques, whether using a tool specifically developed for the purpose, or existing tools such as FDR [12], or converting to other notations suitable for code generation (e.g., converting CSP to B [9]) and then using the code generating capabilities of the B Toolkit.

*3) Advantages of the R2D2C Approach:* We have not yet had an opportunity to apply R2D2C to ANTS, although that is certainly our plan.

In addition to applying it to the HRSM procedures [39], we have applied R2D2C to LOGOS, a NASA prototype Lights-Out Ground Operating System, that exhibits both autonomous and autonomic properties [48], [49]. We illustrate the use of a prototype tool to apply R2D2C to LOGOS in [40], and describe our success with the approach.

Here, we summarize some benefits of using R2D2C, and hence of using Formal Requirements-Based Programming in system development. It is our contention that R2D2C, and other approaches that similarly provide mathematical soundness throughout the development lifecycle, will:

- Dramatically increase assurance of system success by ensuring
    - completeness and consistency of requirements
    - that implementations are true to the requirements
    - that automatically coded systems are bug-free; and that
    - that implementation behavior is as expected
- Decrease costs and schedule impacts of ultra-high dependability systems through automated development
- Decrease re-engineering costs and delays

### *E. Tool Support*

John Rushby [42] argues that tools are not the *most* important thing about formal methods, they are the *only* important thing about formal methods. Although we can sympathize, we do not support such an extreme viewpoint. Formal methods would not be practical without suitable representation notations, proof systems (whether automated and supported by tools, or not), a user community, and evidence of successful application.

We do agree, however, that tool support is vital, and not just for formal methods. Structured design methods *took off* when they were *standardized*, in the guise of UML. But

[1] "Deep" in the sense that the embedding is semantic rather than merely syntactic.

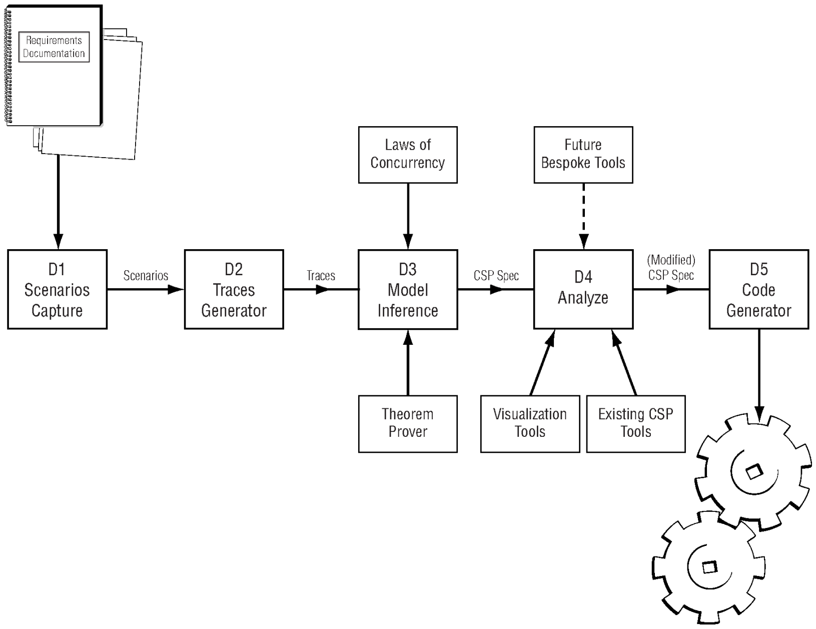


Fig. 4. The entire process with D1 thru D5 illustrating the development approach.

it is only with the advent of tool support for UML that they became popular. The situation is analogous to high-level programming languages: while the community was well convinced of their benefits, it was only with the availability of commercial compilers that they became widely used.

Tools are emerging for the development of complex agent-based systems such as Java-based Aglets and tools for autonomic systems. For automatic code generation and Formal Requirements-Based Programming to be practical, the development community will need commercial-quality tools.

## V. Conclusion

The computing industry thrives on the assumption in the marketplace that software is reliable and correct, but many examples from experience over the decades cast doubt on the validity of this assumption. There is no automated, general purpose method for building correct systems that fully meet all customer requirements. This represents a major gap that has yet to be fully addressed by the software engineering community. Requirements-based programming has been described along with new automated techniques recently devised at NASA for ensuring correctness of the system model with respect to the requirements, as a possible way to close this gap.

In future mission concepts that involve advanced architectures and capabilities — such as swarm missions whose individual elements not only can learn from experience but also must pursue science goals cooperatively — NASA faces system development challenges that cannot be met with techniques currently available in the computing industry. The challenges boil down to building reliability and correctness into mission systems, where complexity, autonomous operation, machine adaptation, dangerous environments, and remoteness combine to push such missions far into uncharted territory in systems engineering. With approaches such as autonomic computing and automated requirements-based programming, NASA will have greater possibilities for achieving success with these advanced mission concepts.

## References

[1] I. Bakam, F. Kordon, C. L. Page, and F. Bousquet. Formalization of a spatialized multiagent model using Coloured Petri Nets for the study of an hunting management system. In *Proc. First International Workshop on Formal Approaches to Agent-Based Systems (FAABS I)*, number 1871 in LNAI, Greenbelt, Maryland, April 2000. Springer.

[2] F. L. Bauer. A trend for the next ten years of software engineering. In H. Freeman and P. M. Lewis, editors, *Software Engineering*, pages 1–23. Academic Press, 1980.

[3] G. Beni and J. Want. Swarm intelligence. In *Proc. Seventh Annual Meeting of the Robotics Society of Japan*, pages 425–428, Tokyo, Japan, 1989. RSJ Press.

[4] E. Bonabeau, G. Théraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine. Self-organization in social insects. *Trends in Ecology and Evolution*, 12:188–193, 1997.

[5] L. Bonnet, G. Florin, L. Duchien, and L. Seinturier. A method for specifying and proving distributed cooperative algorithms. In *Proc. DIMAS-95*, November 1995.

[6] J. P. Bowen and M. G. Hinchey. *High-Integrity System Specification and Design*. FACIT Series. Springer-Verlag, London, UK, 1999.

[7] F. P. Brooks, Jr. No silver bullet: Essence and accidents of software engineering. *IEEE Computer*, 20(4):10–19, April 1987.

[8] R. Büssow, R. Geisler, and M. Klar. Specifying safety-critical embedded systems with Statecharts and Z: A case study. In Astesiano, editor, *Proc. International Conference on Fundamental Approaches to Software Engineering*, number 1382 in LNCS, pages 71–87, Berlin, 1998. Springer-Verlag.

[9] M. J. Butler. *csp2B : A Practical Approach To Combining CSP and B*. Declarative Systems and Software Engineering Group, Department of Electronics and Computer Science, University of Southampton, February 1999.

[10] C. Fellenstein. *On Demand Computing*. IBM Press Series on Information Management. Prentice-Hall, Upper Saddle River, New Jersey, USA, 2005.

[11] C. Fischer. *Combination and Implementation of Processes and Data: from CSP-OZ to Java*. PhD thesis, Universität Oldenburg, Germany, 2000.

[12] Formal Systems (Europe), Ltd. *Failures-Divergences Refinement: User Manual and Tutorial*, 1999.

[13] A. K. Gala and A. D. Baker. Multi-agent communication in JAFMAS. In *Proc. Workshop on Specifying and Implementing Conversation Policies, Third International Conference on Autonomous Agents (Agents '99)*, Seattle, Washington, 1999.

[14] A. J. Galloway and W. J. Stoddart. An operational semantics for ZCCS. In M. Hinchey and S. Liu, editors, *Proc. IEEE International Conference on Formal Engineering Methods (ICFEM-97)*, pages 272–282, Hiroshima, Japan, November 1997. IEEE Computer Society Press, Los Alamitos, Calif.

[15] A. G. Ganek and T. A. Corbi. The dawning of the autonomic computing era. *IBM Systems Journal*, 42(1):5–18, 2003.

[16] J. N. Gray. What next? A few remaining problems in information technology. Turing Award Lecture (ACM FCRC), May 1999.

[17] J. N. Gray. Dependability in the Internet era. In *Proc. High Dependability Computing Consortium Workshop*, Santa Cruz, California, 7 May 2001.

[18] D. Harel. On visual formalisms. *Communications of the ACM*, 31(5):514–530, May 1988.

[19] D. Harel. Biting the silver bullet: Toward a brighter future for system development. *IEEE Computer*, 25(1):8–20, January 1992.

[20] D. Harel. Comments made during presentation at "Formal Approaches to Complex Software Systems" panel session. *ISoLA-04 First International Conference on Leveraging Applications of Formal Methods*, Paphos, Cyprus. 31 October 2004.

[21] M. G. Hinchey and S. A. Jarvis. *Concurrent Systems: Formal Development in CSP*. International Series in Software Engineering. McGraw-Hill International, London, UK, 1995.

[22] M. G. Hinchey, J. L. Rash, and C. A. Rouff. Requirements to design to code: Towards a fully formal approach to automatic code generation. Technical Report TM-2005-212774, NASA Goddard Space Flight Center, Greenbelt, MD, USA, 2004.

[23] M. G. Hinchey, J. L. Rash, and C. A. Rouff. Towards an automated development methodology for dependable systems with application to sensor networks. In *Proc. IEEE Workshop on Information Assurance in Wireless Sensor Networks (WSNIA 2005), Proc. International Performance Computing and Communications Conference (IPCCC-05) (Reprinted in Proc. Real Time in Sweden 2005 (RTiS2005), the 8th Biennial SNART Conference on Real-time Systems, 2005)*, Phoenix, Arizona, 7–9 April 2005. IEEE Computer Society Press, Los Alamitos, Calif.

[24] C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.

[25] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall International Series in Computer Science. Prentice Hall International, Englewood Cliffs, NJ, 1985.

[26] P. Horn. Autonomic computing: IBM's perspective on the state of information technology. Presented at agenda 2001, scotsdale, arizona, 2001, IBM T. J. Watson Laboratory, October 15, 2001.

[27] P. M. Horn. Meeting the needs, realizing the opportunities. In C. W. Wessner, editor, *Capitalizing on New Needs and New Opportunities: Government - Industry Partnerships in Biotechnology and Information Technologies (2001) Board on Science, Technology, and Economic Policy (STEP)*, pages 149–152. The National Academies Press, 2001.

[28] IFAD. The VDM++ toolbox user manual. Technical report, IFAD, 2000.

[29] JPL Special Review Board. Report on the Loss of the Mars Polar Lander and Deep Space 2 missions. Pasadena, California, USA, March 2000.

[30] M. Kaufmann, P. Manolios, and J. Moore. *Computer-Aided Reasoning: An Approach*. Advances in Formal Methods Series. Kluwer Academic Publishers, Boston, 2000.

[31] K. Lano and H. Haughton. *Specification in B: an Introduction Using the B-Toolkit*. Imperial College Press, London, UK, 1996.

[32] H. W. Lawson. Rebirth of the computer industry. *Communications of the ACM*, 45(6):25–29, 2002.

[33] N. G. Leveson. Medical devices: The Therac-25 story. In *Safeware: System Safety and Computers*, pages 515–553. Addison Wesley Publishing Company Inc., 1995.

[34] J. L. Lyons. ARIANE 5: Flight 501 failure, report by the inquiry board, 19 July 1996.

[35] The MathWorks, Inc., Natick, Massachusettes. *Getting Started with MATLAB*, 2000.

[36] D. L. Parnas. Software aspects for strategic defense systems. *American Scientist*, November 1985.

[37] D. L. Parnas. Using mathematical models in the inspection of critical software. In *Applications of Formal Methods*, International Series in Computer Science, pages 17–31. Prentice Hall, Englewood Cliffs, NJ, 1995.

[38] D. Patterson and A. Brown. Recovery-Oriented Computing (Keynote talk). In *Proc. High Performance Transaction Systems Workshop (HPTS)*, October 2001.

[39] J. L. Rash, M. G. Hinchey, C. A. Rouff, and D. Gračanin. Formal requirements-based programming for complex systems. In *Proc. International Conference on Engineering of Complex Computer Systems*, Shanghai, China, 16–20 June 2005. IEEE Computer Society Press, Los Alamitos, Calif.

[40] J. L. Rash, M. G. Hinchey, C. A. Rouff, D. Gračanin, and J. D. Erickson. A tool for requirements-based programming. In *Proc. International Conference on Integrated Design and Process Technology (IDPT 2005)*, Beijing, China, 13–17 June 2005. The Society for Design and Process Science.

[41] C. A. Rouff, W. F. Truszkowski, J. L. Rash, and M. G. Hinchey. A survey of formal methods for intelligent swarms. Technical Report TM-2005-212779, NASA Goddard Space Flight Center, Greenbelt, Maryland, 2005.

[42] J. Rushby. Remarks, panel session on The Future of Formal Methods in Industry. In J. P. Bowen and M. G. Hinchey, editors, *Proc. 9th International Conference of Z Users, LNCS 967*, pages 239–241, Limerick, Ireland, September 1995. Springer-Verlag.

[43] R. Sterritt. Towards autonomic computing: Effective event management. In *Proc. 27th Annual IEEE/NASA Software Engineering Workshop (SEW)*, pages 40–47, Greenbelt, Maryland, USA, 3–5 December 2002. IEEE Computer Society Press, Los Alamitos, Calif.

[44] R. Sterritt. Autonomic computing. *Innovations in Systems and Software Engineering: a NASA Journal*, 1(1), April 2005.

[45] R. Sterritt and D. W. Bustard. Autonomic computing—a means of achieving dependability? In *Proc. IEEE International Conference on the Engineering of Computer Based Systems (ECBS-03)*, pages 247–251, Huntsville, Alabama, USA, April 2003. IEEE Computer Society Press, Los Alamitos, Calif.

[46] R. Sterritt and M. G. Hinchey. Why computer based systems *Should* be autonomic. In *Proc. 12th IEEE International Conference on Engineering of Computer Based Systems (ECBS 2005)*, pages 406–414, Greenbelt, MD, April 2005.

[47] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff. NASA's swarm missions: The challenge of building autonomous software. *IEEE IT Professional*, 6(5):47–52, September/October 2004.

[48] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff. Autonomous and autonomic systems: A paradigm for future space exploration missions. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 36(3):279–291, May 2006.

[49] W. F. Truszkowski, J. L. Rash, C. A. Rouff, and M. G. Hinchey. Some autonomic properties of two legacy multi-agent systems — LOGOS and ACT. In *Proc. 11th IEEE International Conference on Engineering Computer-Based Systems (ECBS), Workshop on Engineering Autonomic Systems (EASe)*, pages 490–498, Brno, Czech Republic, May 2004. IEEE Computer Society Press, Los Alamitos, Calif.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 649–656

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Minos—The design and implementation of an embedded real-time operating system with a perspective of fault tolerance

Thomas Kaegi-Trachsel
Native Systems Group
ETH Zurich
8092 Zurich, Switzerland
Email: thomas.kaegi@inf.ethz.ch

Juerg Gutknecht
Native Systems Group
ETH Zurich
8092 Zurich, Switzerland
Email: gutknecht@inf.ethz.ch

***Abstract*—This paper describes the design and implementation of a small real time operating system (OS) called *Minos* and its application in an onboard active safety project for General Aviation. The focus of the operating system is predictability, stability, safety and simplicity. We introduce fault tolerance aspects in software by the concept of a very fast reboot procedure and by an error correcting flight data memory (FDM). In addition, fault tolerance is supported by custom designed hardware.**

## I. Introduction

WE DEVELOPED Minos in the context of a European Union Research project called *Onbass* [1]. The following quote of the Onbass homepage gives an overview of the project goals:

> The final goal of the project is to design, develop, test and validate an on-board active real-time data processing system that will monitor flight related parameters and react in the case of a proliferation of risk to the aircraft or its occupants. The system will recognise undesirable trends or patterns in data relating to the various aircraft agents (aircraft, systems, pilot) by analyzing and comparing current flight data against previously accumulated aircraft-specific behavioral data. As a result, timely interventions could be made in order to eliminate the associated risk(s) or to minimise the severity of the corresponding effects. In addition, the system will offer invaluable and comprehensive data for post-flight analysis upon which aviation safety bodies could base and/or redesign safety policies and procedures.

The interested reader may refer to [2]–[4] for further information about the application side of the Onbass project, the theory of Active Safety and its implementation.

In this paper we shall describe the operating system developed during the project, with an emphasis on two fault tolerance aspects: a.) recovering from memory faults mainly caused by radiation and b.) reliably recording flight data in a *Flight Data Memory* (FDM). A FDM is a reliable persistent storage system for flight data such as heading, temperature, engine information, etc. recorded in real time during the flight.

The main concept used to increase dependability [5] in our project is managed redundancy. Duplication of the main memory (see chapter II) and of the flight data memory (see chapter VI) lead to a substantially higher level of reliability.

## II. The Hardware Platform

The hardware platform was custom designed and built for this project by IRoC Technology in Grenoble, France, according to the requirements given by the Onbass project specification. The FPGA implementation [6] features a CPU, synthesised from a standard (non fault tolerant) ARM7TDMI IP core by Actel, a fault tolerant main memory (RAM) and a fault tolerant ROM.

Fault tolerant RAM and ROM provide safeguards against both temporary and permanent errors. Temporary errors, for example bit flips, are mainly induced by radiation such as alpha particles, neutrons or heavy ions that are generated by solar winds or by other cosmic radiation. At sea level, these events happen rarely because most of the radiation is filtered by the earths atmosphere. However, at typical flight altitude (10km above sea level) or in deep space several hundred or thousand kilometers above ground, experiments have shown [7] that such events occur as frequently as 5.55 times per megabyte RAM per day in average. ROM is much less susceptible to such events, but they can still occur.

Permanent errors on the other hand affect both, RAM and ROM equally. Such errors usually manifest themselves as failures of parts of or the entire physical memory chip. In the former case, only certain regions are affected, in the latter case the whole chip fails.

If a temporary fault occurs, the system should recover as quickly as possible and continue its operation from the most recent consistent state. In the case of a permanent error, the system is supposed to still continue its operation, possibly in a degraded mode, after signalling the failure to the runtime and application.

The following sections describe the strategies chosen for dealing with the two types of error just described.

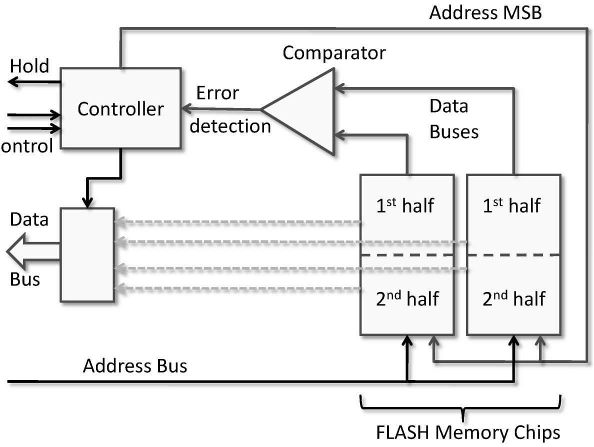


Fig. 1. ROM Implementation (Picture courtesy of IRoC Technology)

**Memory Subsystem:** As a first precaution, static RAM was chosen instead of dynamic RAM as it is faster and more resilient to temporary errors. Physical duplication of memory chips combined with a CRC mechanism provides immunity against a complete failure of any single memory chip. Read/ write operations are always performed simultaneously on two memory chips and, if one of them fails, the error is immediately flagged to the OS.

Furthermore, a word-based error detection mechanism was implemented. A 36-bit wide version of SRAM was chosen, where the 4 extra bits per word are used to store a hardware-generated *Cyclic Redundancy Check* (CRC). At each read operation, the data of both memory chips involved is compared against each other. If the comparison fails, the CRC is used to determine the faulty chip, and its partner chip is used to automatically correct the faulty memory location. As these operations are integrated into the memory controller, the comparison can be done without performance penalty, and the correction in case of a mismatch requires just one additional memory access (one CPU cycle). Counters of all corrected and uncorrected errors are provided to the runtime for further analysis or logging.

Triplicated memory was considered an alternative to the duplication but an unimproved level of protection regarding temporary faults, longer Mean Time To Failure (MTTF), lower hardware costs and lower heat dissipation favoured the chosen approach.

**Flash Memory:** The binary image of the OS is stored in flash-memory (ROM). The system ROM is physically duplicated but, as it is less susceptible to external influences, no extra bits for error detection are provided. Instead, two instances of the OS image are stored on each ROM chip, one in the first half of the address space and one (in reverse bit order) in the second half, as shown in Figure 1. This procedure can be justified in our case by the extremely small size of the OS image.

In order to understand how error detection/ correction in ROM works, we first remember that ROM chips are actually 16 bit wide[1], so that two corresponding 16 bit entities from the two ROM chips fit in one 32 bit word. At boot time, the hardware controller uses this fact for error detection when copying the binary OS image wordwise from ROM into RAM. If the two 16 bit entities in a word do not match, the image stored in the lower ROM part is considered corrupted, and the correct data has to be retrieved from the upper ROM part. As this procedure is only initiated once at boot up, the additional overhead is negligible.

Calculations [6] showed that the system just described not only features up to 99% mitigation efficiency regarding to non-permanent (transient) errors but it also enjoys twice as long Maintainance Free Operating Periods (MFOPs) if compared with a non-redundant implementation. Both, a more extensive reliability analysis and more implementation details, are given in [6].

## III. The Programming Language Oberon

The programming language used in the Onbass project is *Oberon 07* [8], [9], a modular descendant of Pascal and Modula-2. Oberon 07 is a simple and safe variant of *Oberon* [10] for embedded systems. For example, the option of in-line assembler code has been removed completely from the language and replaced with a set of safer and more structured custom built-in functions [8]. Oberon07 also provides a mechanism for accelerated calls and execution of ”leaf” procedures that do not contain (further) procedure calls. Parameters and local variables in leaf procedures are allocated in registers (instead of on the stack) whenever possible. The Oberon language is especially suitable for safety critical applications as it is completely type safe and allows no unsafe operations such as type conversions, etc. except in explicitly marked sections for kernel code.

## IV. The Applications

Before explaining the design of the Operating System, we would like to give a short overview of the Onbass applications that were supposed to run on the system. This gives an idea about the requirements:

**Flight Data Acquisition** The Onbass system is installed in a general aviation airplane such as a Piper Lance and directly connected to the onboard air data computer that delivers a set of sensor data to the Onbass system up to eight times per second.

**Black Box Recording** After the data acquisition, the raw black box data is stored in the flight data memory (see chapter VI) for later analysis or recovery.

**Data Parsing** Next, the acquired data is parsed and validated according to the air data computer specification.

**Flight Mode Detection** The current flight mode (take off, cruise, etc) is then evaluated. This is especially important, as the valid airplane constraints such as speed, vertical speed, etc. heavily depend on the flight mode.

[1] ROM chips use an interlaced addressing scheme, where the first 16 bits in address space belong to the first chip, the second 16 bit belong to the second chip, the third 16 bit to the first chip again, etc.

**Airplane safety checks** A set of rules is applied to the current airplane flight state to validate the operation of the airplane in terms of safety. In case of detected deviations, a warning is displayed to user via the web interface.

**Web Server** A web server is used to display system information such as configuration options, warnings, etc. via a separate computer to the pilot.

**Replay** For post flight analysis, the system can be configured to load the flight data from the FDM and and use it as data input for the application instead of acquiring the data from the air data computer.

**Supportive tasks** Various other tasks are required to support the system such as additional logging, polling driver tasks for UART and MMC etc.

The whole flight data acquisition and analysis must obviously be performed in real time and must be finished before the next flight data set arrives. This imposes requirements on the real time capabilities of the system.

## V. The Design of Minos

As a starting point for the runtime system we chose *HelyOS* [11], an embedded operating system for the control of autonomously operating model helicopters developed at ETH Zurich.

We did a substantial rewriting of HelyOS and customised it towards our specific needs as it was too limiting. The resulting system is called *Minos* and enjoys the following qualities:

- Very small, simple and efficient
- Suitable for safety critical applications
- Predictable in terms of task execution time
- Easily portable to other platforms
- Highly configurable at boot time
- Fast boot up time

In the next few chapters we shall give a short introduction to the key concepts of the Minos design.

### A. Fast Boot Up

Special attention was given to system boot up time. A hardware watchdog is used to detect ”stuck” programs caused by a malfunction of either hardware or software. As soon as the watchdog detects a timed-out activity of any kind, the OS and the applications are restarted and brought into the most recent consistent state. This is quite easily possible in this case because all the relevant data has by been stored in the flight data memory by concept IV. In order to achieve a minimum downtime it is important to implement an ultra fast boot up mechanism that, in our case, takes less than 0.5 seconds. Unavoidably this requirement has an impact on various subsystems such as the flight data memory (see chapter VI).

### B. Memory Management

Sytems of the Oberon family [12], [13] traditionally use a completely type-safe ”managed” runtime including garbage collection. However, it is widely known that garbage collection introduces unpredictable latencies in the execution of application programs and garbage collectors for real time systems are complex and difficult to design and implement. As a consequence, other traditional garbage collector based systems, i.e. Java, propose in their real time variants (Real Time Java [14]) the addition of memory outside the scope of the garbage collector for time critical software parts.

As explicit memory allocation and deallocation is inherently unsafe and therefore incompatible with safety-critical applications, the only viable option is not to generate garbage at all. We use a closed system approach that still allows applications to allocate dynamic memory but at initialisation time only. An additional benefit of closed systems is that they can never run out of memory.

Unavoidably however, there is a price to pay for such simplicity. It is the need for compensatory support for application programming. For example, it is impossible for Minos to permanently keep a dynamic metadata structure for files in memory. Instead, metadata needs to be consistently stored in flash memory, and sector caches are taken from a preallocated buffer pool. As the pool is only used internally by the filesystem and as blocks are automatically recycled when the buffer pool is empty, no ”out of memory” situations can ever occur.

### C. Interrupt Handling

The interrupt handling scheme in Minos is again kept simple. At boot time, the kernel installs a single, general interrupt handler that is responsible for dispatching all signalled interrupts. Device drivers register their own handler in the kernel, where only one handler per interrupt source is currently supported. Interrupt handlers are non interruptable, and their processing time must be kept in limits in order not to compromise realtime guarantees. In the case of multiple interrupts pending, the kernel calls the handlers in the order of ascending interrupt numbers.

### D. Task Model

*Original Tasking Scheme:*
HelyOS uses an ingeniously simple, preemptive tasking scheme that is characterised by the following principles. First, the scheme distinguishes four priorities corresponding to four different task types: Interrupt handlers, high priority periodic tasks running at period $s$, low priority periodic tasks running at period $l = k * s$ with fixed $k$, and background tasks.

Second, HelyOS uses the following scheduling policy:

- Interrupt handlers have highest priority and preempt all other tasks.
- High priority periodic tasks preempt low priority tasks and background tasks but no interrupts.
- Low priority periodic tasks preempt background tasks but no others.
- Background tasks do not preempt any tasks.

An interesting consequence of this scheduling policy is the fact that each task must run to completion before any other task of the same priority can start its execution, with the immensely beneficial implication that a single stack is sufficient in principle for implementing the entire scheduling scheme.

*Modified Tasking Scheme:*
The scheduling policy just described is not powerful enough for our application. In particular, the restriction to merely two types of periodical tasks corresponding to two fixed periods is too rigid. However, in the interest of avoiding the full complexity of managing multiple stacks and of mastering an intricate synchronisation mechanism, we refrained from switching to a fully general model. Instead, we generalised the HelyOS model appropriately. The most important modification is a new strategy oriented towards ”earliest deadline” scheduling. Both a period and a priority number are preassigned to each task, where the period corresponds to the ”earliest deadline” and the priority number is used to resolve ties. A priority number is also preassigned to background tasks, but of no period of course.

Minos Scheduling Principles:

- Interrupt handlers have highest priority and preempt all other tasks.
- Periodic tasks are scheduled according to their deadline as derived from their period. If two tasks have the same period, the execution order is defined by their priority.
- Background tasks are scheduled according to their priority.
- Tasks can only be preempted by tasks with a shorter deadline.

A consequence of this scheme is the fact that the use of periodic tasks for polling external events is inappropriate, and that interrupts must be used instead. The reason is that a delay in the order of the period (currently 5 ms, but this could be easily changed) is often unacceptable. However, this scheme is still suitable in our case because we are not primarily interested in very fast reaction times but in a predictable behavior in terms of both time and order of execution.

If a deadline was missed, then an optional delegate provided by the task object is called. The delegate is responsible for taking recovery actions such as, in the simplest case, merely logging the problem. The next execution of this task is then skipped to give the system time to recover. Note that such a behavior is also necessary to prevent possible stack overflows.

Another nice consequence of our simple tasking model is that accessing shared data often needs no synchronisation as the tasks are serialised implicitly. This is notably the case if data structures are shared among tasks of the same period and background tasks only. In the (rarely occurring) other cases where a locking mechanism is required, we use a global system lock that simply disables all interrupts (including timer interrupt).

Figure 2 illustrates this tasking scheme. After background task A runs to completion, task B is automatically executed. At this time, neither a periodic task nor an interrupt is pending. At time 50, periodic tasks P and Q are both due, whereas Q has the smaller period and therefore shorter deadline than P. The task B is preempted and Q executed. When Q finishes, P is automatically invoked because periodic tasks have higher priority than background tasks. At time 63, an interrupt is signaled by the hardware and the respective interrupt handler is called immediately. P is resumed as soon as the interrupt handler has finished its task. When P finishes executing, neither a periodic task nor an interrupt is pending and background task B is resumed.

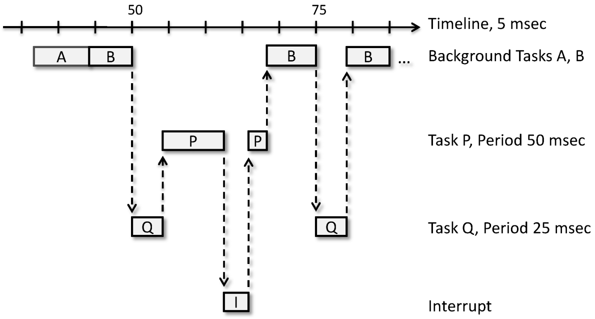


Fig. 2. Scheduling example

A question naturally arising here is if priority inversion is possible that is if a scenario can be found where some high priority task needs access to a shared resource that is locked by a low priority task so that intervening medium priority tasks can effectively block the execution of the high priority task. In our tasking model, the only synchronisation primitive provided is the global lock. When a task acquires this lock, the task is implicitly set to highest priority (priority ceiling) and the scheduling mechanism is disabled while the lock is held. Because it is impossible that any task holding the global lock is interrupted by another task, priority inversion is impossible.

The scheduler itself runs in linear time (linear in the number $O(n)$ of tasks as the due time has to be calculated for each task), and it is thus easy to calculate an upper bound for the scheduler execution time.

Our tasking model (see section V-E) pays out in a very efficient task switching algorithm. In fact, the switch from the task scheduler to any other task is synchronous and amounts to just a procedure call (delegate), and the return to the scheduler simply corresponds to the return from the procedure. Only interrupts are asynchronous and therefore require saving of registers on the stack.

### E. *Stack Management*

The stack management is equally simple. We use a fixed number of separate stacks, one for interrupts, one for periodic tasks and one for background tasks respectively. In principle, one stack would suffice because each task either runs to completion or is preempted by a task of a higher priority, which in turn runs to completion, so that each preempted task finds a clean stack when resumed. However, using a fixed number of separate stacks simplifies the handling of traps.

### F. *Boot Configuration Procedure*

A particular requirement in our project specification is full configurability of the system at boot time. In the interest of readability, flexibility and ease of configuration, we chose an

XML [15] approach. For each hardware component and each software component, a separate XML section is provided, and a complete set of default settings for all core components is hardcoded into the program and activated at run time before the XML configuration parser is invoked. This serves the purpose of putting the system into a consistent working state even before the configuration file has been read.

Due to the restricted policy of allocating dynamic memory, we implemented a considerably simplified SAX [16] based parser that itself does not rely on heap memory. As the system must be able to operate independently of any external host computer, the configuration file can alternatively be stored in flash memory in the device itself or downloaded from a host terminal at boot time.

The initialization procedure resulting from all these constraints looks like this:

1) Kernel initialisation, platform setup.
2) Hardware configuration and initialisation by default settings.
3) Mounting of RAM and ROM disc.
4) Acquisition of XML configuration file either by loading it from the ROM disk or by downloading it via a serial connection from a host computer.
5) Processing of the ”autostart” section in the XML file. Can execute any arbitrary command but is especially used to register XML handler plug-ins for the configuration process.
6) The XML parser scans through the rest of the XML file and calls the appropriate plug-ins if one is registered.
7) Enter main command loop.

The configuration scheme described above proved to be extremely powerful and flexible. The only negative aspect is the strict top-down parsing order imposed by SAX, which sometimes leads to clumsy configuration clauses.

### G. Modular System Structure

As shown in figure 3, Minos is a fully modular and hierarchically structured system. For the sake of better readability, the (optional) boot configuration mechanism and the XML parser are omitted in the figure. The RAM disk is modelled as a *volume object* for filesystem containers. Other examples of volume objects are ROM disks and Flash disks. Again in the interest of readability, the dependencies on modules *Log* and *SerialLog* are also omitted.

**SYSTEM** This is a pseudo module provided by the Oberon compiler; it provides potentially unsafe functionality required for low level system programming such as memory mapped input/ output. Utmost care must be exercised in code that uses features from module SYSTEM because such code must be considered as potentially unsafe.

**Platform** Platform specific information such as memory layout, interrupt numbers and memory mapped I/O registers. By merely replacing the implementation of this module, Minos can be adapted to a variety of processors of the same architecture, including for example the Marvell PXA255 and the Marvell PXA270.

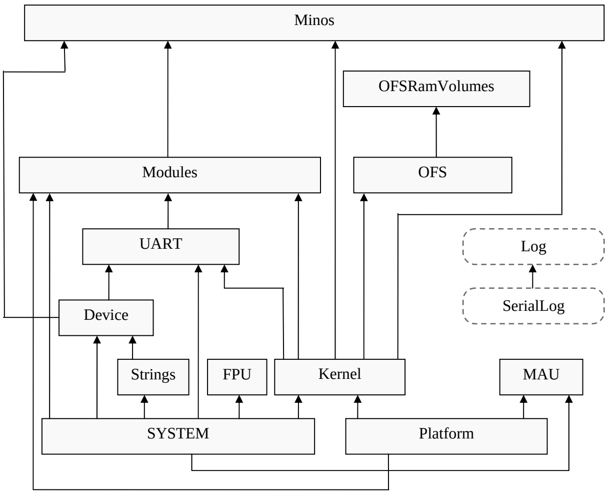


Fig. 3. Core system modules

**MAU** Memory Allocation Unit, provides the implementation of the memory allocation logic. This module is referenced by the compiler and should not be used directly.

**FPU** Floating Point Emulation. This Module implements runtime support for basic Math operations on floating point numbers as well as for integer division. It is used by the compiler rather than by applications.

**Strings** Basic functionality for copy, search, add, etc. operations on strings. This module is added to the kernel for reuse to avoid code duplication.

**Kernel** The Kernel provides platform-specific tasks such as system initialisation, interrupt handling, timers, etc. It is highly unportable and must be adapted to every platform individually.

**Device** An abstract Character Device used as an abstract interface by plug-in device drivers. It allows the dynamic addition or change of input/ output devices such as (real or virtual) serial ports at runtime.

**Uart** UART device driver, implements a Device.Device plug-in object.

**Log** An abstract logging device that can be used to display log output on different devices such as serial port or Web browser.

**SerialLog** Log over the serial connection. A concrete implementation of module Log.

**OFS** Oberon File System. Provides file operations such as creating, deleting, reading or writing files. It also implements the Oberon File System that is based on the notion of volumes, where a volume is an abstract file system container that provides read/ write access to blocks of fixed size.

**OFSRamVolumes** RAM Disk support. Implements a volume declared as an abstract object in OFS

**Modules** Dynamic module loader. Allows to dynamically download, link and execute modules at runtime.

**Minos** Implements the scheduler and the trap handler and offers user interface commands to be activated via a remote terminal.

The set of modules presented here is a basic and self-contained subset of all Minos modules. The modular concept allows software developers to seamlesly add new functionality to the system at any time by simply linking the appropriate modules to the current image. Minos also allows modules to be downloaded, linked and executed dynamically at runtime. This is very convenient for prototyping, testing and debugging. For example, a flexible testing environment can be built by merely flashing to ROM a version of the basic Minos runtime that automatically downloads application code at boot time.

The size of the full operating system including all the above listed modules, the XML configuration parser and the boot configuration mechanism is ca. 100 Kbytes. This is less than half the size of a comparable commercial system such as, for example, VxWorks by Wind River (Size of VxWorks 6.2 without XML parser is about 250 Kbytes Basic OS profile [17]).

## VI. Flight Data Memory

### A. Introduction

An avionics Flight Data Memory (FDM), also called a ”black box”, is an extremely robust and reliable flight data recorder that is typically used in aircrafts for post flight/ post disaster analysis. The current trend in General Avionics goes towards declaring FDM mandatory even in small aircrafts [18]. FDMs often get their input streamed down from an air data computer (ADC) that, in turn, collects the data from a variety of sensors across the airplane. Optionally, FDMs can also be connected to other input sources.

Streams of sensor data like heading, height, fuel flow, etc. must be recorded reliably on long life and non-volatile medium such as flash memory or special magnetic tape [18]. Data redundancy schemes such as CRC-32 [19] for error detection, or Reed Solomon for error detection and correction are commonly used to further enhance the reliability of the FDM. Our own choice in Onbass was using CRC-32 for error detection and data duplication on two separate multimedia cards (MMC) for error correction.

The FDM in Onbass was designed with the following requirements in mind:

- Simple design
- Fast data storage and retrieval
- Minimum 15 years life time
- Fast recovery after unexpected reboot & transparent flight resuming
- Transparent MMC device recovery in case of faulty read or write operation
- Small memory footprint
- Support for replaying stored flights
- Human decipherable format
- Space efficiency

### B. MMC vs Compact Flash

We had to decide between multimedia cards (MMC) [20] and compact flash cards (CF) [21]. MMC has the advantage of physical compactness and of a low pin count (7 pins), whereas CF comes with a built in wear-leveling algorithm but has a high pin count (50 pins). We decided in favor of a low pin count because physical connections are arguably the most critical components in any system from a reliability point of view.

### C. Flash Properties and Limitations

Flash memory is organised in blocks (usually 512 bytes) and in *erase units* consisting of a number of adjacent blocks (usually 32 or 64 [22]. Reads and writes are performed blockwise, and writes must be made only to previously erased blocks. Blocks can either be erased automatically or manually, where the former option is more comfortable while the latter is faster. The number of erases per unit before ”wearing out” is limited, typically to some number between 10000 and 100000. In the interest of longevity of the flash card, the use of a ”wear leveling” strategy is highly advisable. Wear-leveling means that erase cycles and write cycles are evenly distributed across the memory chip. Traditional file systems are unsuitable for flash cards because they exhibit *hot spots* such as meta data fields that are frequently updated.

### D. Analysis and Design Considerations

In Onbass we can take advantage of the fact that the size of records to be stored is fixed (flight data frame plus warnings). As mentioned above, the use of a standard file system is unsuitable as it typically exhibits hot spots. Much of the research [23] into overcoming this problem introduces some virtual-to-physical block number mapping. Two kinds of data structures are typically suggested for this purpose. *Direct maps* map a logical block number (index $i$) to its physical sector number. Unfortunately, such data structures have typically a footprint in the order of several megabytes [24]. *Inverse maps* store in location $i$ the virtual block number corresponding to sector $i$. These maps are usually stored on the flash disk itself and are mainly used for regenerating the direct map at boot time. However, after an unexpected reboot, it would take considerable time to rebuild the direct and indirect maps, which is incompatible with the request of a fast reboot time. Also, many of these algorithms are patented.

While algorithms based on virtual block mapping can greatly extend the lifetime of flash memory, this comes at the price of increased complexity, of a large memory footprint, and of a garbage collection mechanism for reclaiming invalid sectors. As this is again incompatible with realtime constraints, it is not an option in our case.

Another approach is the use of a log structured file system such as *JFFS* [25]. Log structured file systems do not structurally separate metadata and payload data but instead maintain a comprehensive log of all performed operations in chronological order. While wear leveling is implicit in such systems, they still suffer from the garbage collection problem, which again disqualifies them for the use in our project.

We should also remember that one of the Onbass requirements (see chapter VI-A) is readability of the data recorded in an FDM without the help of a software decoder. A FDM must

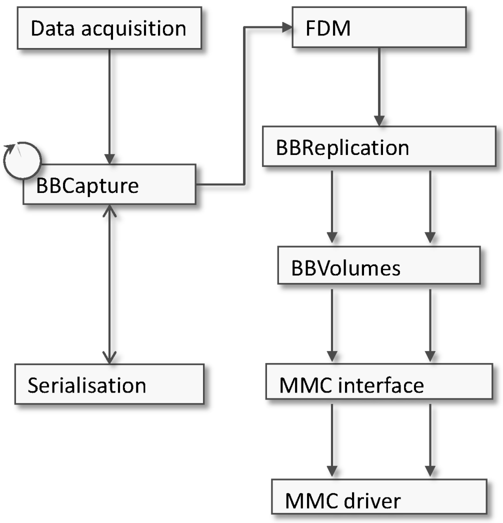


Fig. 4. FDM Implementation Overview

by law be fully recoverable from scratch. This requirement de facto excludes any sophisticated allocation scheme because recovering data without decoder software would either be impossible at all or at least take considerable efforts.

Therefore, we refrained from using such advanced storing schemata and decided in favor of a simple circular buffer structure, where each flight data record occupies the same number of flash card blocks. Whenever a new erase unit is entered, an erase operation is performed as a preparation for subsequent writing. The obvious drawback of this scheme is internal fragmentation if the size of a data element is not an exact multiple of the elementary block size. We considered this as acceptable in particular because more sensor data will have to be stored in the future, which reduces the fragmentation overhead.

### E. *Implementation*

In addition to the actual flight data, some metadata is recorded on the FDM: The FDM header occupies one erase block and describes the current contents of the FDM. It contains a fingerprint, the number of flights currently stored in the FDM, the flight numbers of the oldest and newest flights currently stored in the FDM, and a list of flight indexes. A flight index in turn points to the first and the last header block of the corresponding flight. Flight header blocks again contain a fingerprint (for the support of a scavenging process), the flight number, the starting time of the flight and the date.

During flight recording, the pointer to the last block of the current flight index is declared invalid before the flight has properly been closed. This allows the system to detect unexpected reboots and, as each flight data block is stamped with the corresponding flight number, to use a binary search procedure for locating the most recently recorded flight data block.

Prior to writing, the FDM software must check whether the next sector is free or the start of the oldest flight in the circular buffer. In the latter case, the FDM software deletes the oldest flight and continues recording.

### F. *Module View*

Figure 4 shows the implementation of the FDM as a layered modular system.

**Flight Data Memory (FDM)** This module provides an API for starting and ending the recording of flight data, for storing and retrieving flight data frames and for performing other administrative tasks.

The standard procedure to initialise the flight data memory is registering the FDM module with the XML configuration mechanism by calling the *Install* procedure and then configuring the flight data memory according to the specification in the XML configuration file. A replay mode (replay of a stored flight) can also be enabled via the configuration file.

**BBCapture** This module is responsible for acquiring the flight data from the application and for periodically storing it. For this purpose, *BBCapture* installs a periodic task.

**Serialisation** The *Serialisation* module is responsible for serialising flight data frames into a contiguous data stream to be fed to the FDM.

**BBReplication** This layer partly implements the error detection/correction algorithms. In detail, the module is responsible for computing the CRC-32 for each block. The CRC-32 is automatically generated during write operations and automatically checked during read operations. All read/ write operations are performed sequentially on two configurable partitions on two distinct flash cards. At each read operation, data integrity is checked automatically. If a faulty CRC-32 is detected, the healthy copy is used to fix the data by merely rewriting the faulty block and an error indication is returned. If one of the flash cards fails permanently, the system still continues to record data on the healthy card, and a log message plus an appropriate status code are generated to indicate the failure.

**BBVolumes** Flash disks usually come with a standard partition table and thereby support the coexistence of an FDM and standard file systems on the same disk. *BBVolumes* implements a disk volume object that represents a logical volume, in our case a partition, and extends its functionality with the ability of erasing erase units on the flash disk.

In case of a malfunctioning MMC controller or card, both, the controller and the cards are automatically reset and the failing operation is retried. If it fails again, an error code is returned to indicate the failure.

**MMC Interface / MMC Driver** These two modules implement the multimedia card driver, an interface for reading, writing and erasing blocks and some administrative support such as acquiring cards.

## VII. Conclusion and Future Work

We have built a small and highly reliable realtime operating system that is targeted at safety-critical applications such as the onboard monitoring purpose specified by the Onbass project specification. In numerous real and simulated flight trials (with simulated hazards), the system has proved to operate correctly and reliably. However, some scenarios pushed the system to its limits, especially the MMC subsystem. Because recording of flight data is of prime importance in Onbass, it is performed by a periodic real time task. This is possible because writing a disk sector usually takes less than 500 usec. However, in case of any failure, the MMC specification defines a default timeout value of 250 msec [22], which easily goes beyond the time limit of the corresponding periodic task. Our system proved to work reliably even in such cases but only thanks to the low system load. It is advisable to extend the tasking mechanism by an option of suspending a task while waiting for some hardware event. Alternatively, accesses to the MMC controller could be encapsulated in a separate periodically polling task. However, this would lead to a degradation of the sequential read/ write performance as the maximum throughput would be limited by the minimum polling period of the task.

The FDM has proved to work reliably as well. Tests performed by intentional modifications of the stored flight data on one or both of the MMC cards showed that all tested inconsistencies (data errors) are reliably detected and (where possible) fixed. A potential improvement in terms of wear leveling could be achieved by periodically moving the FDM header (which is a hot spot) across the medium. Adding spare sectors or entire erase units for a potential replacement of blocks with permanent errors could also improve the lifetime of the system.

## ACKNOWLEDGMENT

The authors would like to thank Felix Friedrich and Florian Negele for their help in designing and implementing the system and for the many hours of intense collaboration and discussions. Many thanks also go to Brian Kirk and to Igor Schagaev for their inspiration and all the contiguous critical and constructive discussions.

## REFERENCES

[1] Onbass consortium, “Onbass website,” http://www.onbass.org, 2007.

[2] Onbass consortium, “Onbass D1.2 pass functional&reliability-models,” Onbass consortium, Tech. Rep., 2007.

[3] I. Schagaev, B. Kirk, and V. Bukov, “Applying the principle of active safety to aviation,” EUCASS 2nd European Conference for Aerospace Sciences, Tech. Rep., 2007.

[4] V. Bukov, V. Chernyshov, B. Kirk, and I. Schagaev, “Principle of active system safety for aviation: Challenges, supportive theory, implementation, application and future,” ASTEC’07 ”New challenges in aeronautics”, Tech. Rep., August 19-23, Moscow, 2007.

[5] A. Avizienis, J.-C. Laprie, and B. Randell, “Fundamental concepts of computer system dependability,” IARP/IEEE-RAS Workshop on Robot Dependability, Tech. Rep., 2001.

[6] D. Alexandrescu, “Onbass deliverable 4.1: Hardware architecture definition,” IRoC Technologies, Tech. Report, 2006.

[7] P. P. Shirvani, “Cots technology & issues-space environments,” Center for Reliable Computing, Stanford University, Tech. Rep., 2003.

[8] N. Wirth, “Oberon-SA, language and compiler,” ETH Zurich, Tech. Rep., 2007.

[9] N. Wirth, “An Oberon Compiler for the ARM Processor,” ETH Zurich, Tech. Rep., 2008.

[10] N. Wirth, “Oberon language report,” ETH Zurich, Tech. Rep., 1990.

[11] M. Sanvido, “A computer system for model helicopter flight control,” ETH Zurich, Tech. Rep., 1999.

[12] N. Wirth and J. Gutknecht, *Project Oberon*, 2005th ed., 2005.

[13] P. J. Muller, “The active object system—design and multiprocessor implementation,” Ph.D. dissertation, ETH Zurich, 2002.

[14] G. Bollella, P. Dibble, and et al., “JSR 1: Real-time specification for java,” RTSJ Technical Interpretation Committee, Tech. Rep., 2006.

[15] C. M. S.-M. E. M. F. Y. e. a. Time Bray, Jean Paoli, “Extensible markup language (xml) 1.0 (fourth edition),” W3C, Tech. Rep., 2006.

[16] W. S. Means and M. A. Bodie, *The Book of SAX*. No Starch Press, 2002.

[17] Wind River Systems, “Wind River General Purpose Platform, VxWorks Edition 3.6,” Wind River Systems, Inc, Tech. Rep., 2007.

[18] Onbass consortium, “Onbass D1.1 application domain definition,” Onbass consortium, Tech. Rep., March 2005.

[19] M. S. et al., “Reversing CRC—theory and practice,” HU Berlin, Tech. Rep., May 2006.

[20] SanDisk, “Multimediacard product manual,” SanDisk, Tech. Rep., 2001.

[21] C. F. Association, “CF+ and compact flash specification revision 4.1,” Compact Flash Association, Tech. Rep., 2007.

[22] SanDisk, “Host design considerations: NAND MMC and SD-based products,” SanDisk, Tech. Rep., 2002.

[23] E. Gal and S. Toledo, “Algorithms and data structures for flash memories,” Tel-Aviv University, Tech. Rep., 2005.

[24] L. Chang and T. Kuo, “An efficient management scheme for largescale flashmemory storage systems,” National Taiwan University, Taipei, Taiwan 106, Tech. Rep., 2004.

[25] D. Woodhouse, “Jffs : The journalling flash file system,” Red Hat, Inc., Tech. Rep., 2001.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Simulator Generation Using an Automaton Based Pipeline Model for Timing Analysis

Rola Kassem, Mikaël Briday, Jean-Luc Béchennec, and Yvon Trinquet
IRCCyN, UMR CNRS 6597
1, rue de la Noë – BP92101
44321 Nantes Cedex 3 – France
firstname.name@irccyn.ec-nantes.fr

Guillaume Savaton
ESEO
4, rue Merlet de la Boulaye – BP30926
49009 Angers Cedex 01 – France
guillaume.savaton@eseo.fr

***Abstract*—Hardware simulation is an important part of the design of embedded and/or real-time systems. It can be used to compute the Worst Case Execution Time (WCET) and to provide a mean to run software when final hardware is not yet available. Building a simulator is a long and difficult task, especially when the architecture of processor is complex. This task can be alleviated by using a Hardware Architecture Description Language and generating the simulator. In this article we focus on a technique to generate an automata based simulator from the description of the pipeline. The description is transformed into an automaton and a set of resources which, in turn, are transformed into a simulator. The goal is to obtain a cycle-accurate simulator to verify timing characteristics of embedded real-time systems. An experiment compares an Instruction Set Simulator with and without the automaton based cycle-accurate simulator.**

## I. Introduction

SIMULATION of the hardware platform takes place in the final stage of development. It can be used for 2 tasks. The first task is the evaluation of the Worst Case Execution Time (WCET) to compute the schedulability of the application [9]. The second task is the test of the application using scenarios before final testing on the real hardware platform. This test is useful because simulation allows an easy analysis of the execution. In both cases, a cycle accurate model of the hardware platform must be used to insure that timings of the simulation are as close as possible to timings of the execution on the real platform.

A common approach for the hardware modelling [5] is based on an hardware centric view; in this approach, the processor is usually modelled by a set of functional blocks. The blocks communicate and synchronise with each other in order to handle the pipeline hazards. A pipeline can be modelled with this approach by designing a functional block for each stage of the pipeline (a SystemC [8] module for instance). This approach is very useful in the design process as it allows synthesis generation. However, simulators that are generated from these kind of models are slow because of the block synchronisation cost. In our approach, we propose to use an Architecture Description Language (ADL) to describe the pipeline and the instruction set of the target architecture. The goal of the pipeline description is to focus on the effects of dependencies and device usage on the timing. This description is transformed into a finite state automaton, which is then transformed into the simulator source code by adding the instruction behaviour (see figure 1). The aim of the automaton is to provide a faster—yet accurate—simulator because hazards of the pipeline are not computed at execution time but at generation time instead. This paper focuses on pipeline modelling and does present the pipeline description in our ADL: HARMLESS [7] (Hardware ARchitecture Modelling Language for Embedded Software Simulation) but the method presented may be extended to include the modelling of other parts of the architecture like branch prediction or cache memories.

## II. Related work

Pipeline and resource scheduling have been studied widely in instruction schedulers, that are used by compilers to exploit the instruction level parallelism and to minimise the program execution time. In [12], Müller proposes to use one or more finite state automata to model the pipeline and build a simulator. Then the simulator is used by the instruction scheduler to compute the execution time of instruction sequences. The automata can be quite large and minimisation techniques may be used to alleviate them. In [13], Proebsting and Fraser use the same approach but a different algorithm which produces directly a minimal automaton. In [1], Bala and Rubin improve the algorithm on [12] and [13] by allowing to replace instructions in already scheduled sequences and present the BuildForwardFSA algorithm which is used to build the automaton. In these works, only structural hazards are taken into account. It makes sense for instruction schedulers.

Other works have been done to build simulator for WCET analysis using an ADL. In [10], Li and al. use the EXPRESSION ADL to build an *executions graph* and express pipeline hazards. The hazards are resolved at run-time. In [14], Tavares and al. use a minimal ADL to generate a model based on Proebsting and Fraser work.

In [2], another kind of pipeline modelling using coloured Petri Nets is presented, but functional behaviour is not taken into account. The simulator cannot execute a real binary code.

In the work presented hereafter, we extend the BuildForwardFSA algorithm to take into account the data hazards (data

This work is supported by ANR (French National Research Agency) under Grant Number ADEME-05-03-C0121.

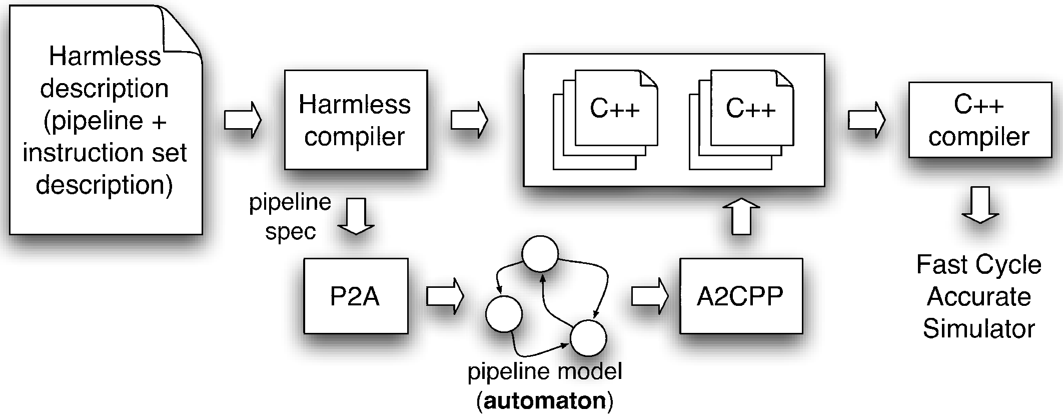

Harmless description (pipeline + instruction set description)
Harmless compiler
C++
C++
C++ compiler
pipeline spec
P2A
pipeline model (**automaton**)
A2CPP
Fast Cycle Accurate Simulator

Fig. 1. Development chain. Tools presented in this paper include *p2a* and *a2cpp* to transform a pipeline description into a fast simulator, using an automaton model.

dependency), the control hazards and the resources which can be held by external hardware devices that are not modelled using one automaton.

The paper is organised as follow: Section III presents how the pipeline is modelled. The two kind of resources and their usage are described. Instruction classes are introduced. Section IV explains the automaton generation algorithm. In section V, a brief description of HARMLESS ADL with a focus on pipeline description, is given. Section VI presents examples. Section VII presents the simulation result provided by our approach. Section VIII explains the relation between simulation and WCET. At last, section IX concludes this paper.

## III. Pipeline modelling

Sequential pipelines are considered in this paper (i.e. there are no pipelines working in parallel nor forking pipelines). An automaton is used to model the pipeline behaviour where a state of the automaton represents the pipeline state at a particular time (see figure 2).

At each clock cycle, the pipeline goes from one state to another according to hazards. They are classified into three categories:

- *Structural hazards* are the result of a lack of hardware resource;
- *Data hazards* are the result of a data dependancy between instructions;
- *Control hazards* that occurs when a branch is taken in the program.

Control hazard are resolved in the simulator at runtime: instructions that are in the delay slot of a branch instruction are dynamically replaced by NOP instructions, if the branch is found to be taken. Constraints resulting from structural and data hazards are used to generate this automaton and modelled using *resources*.

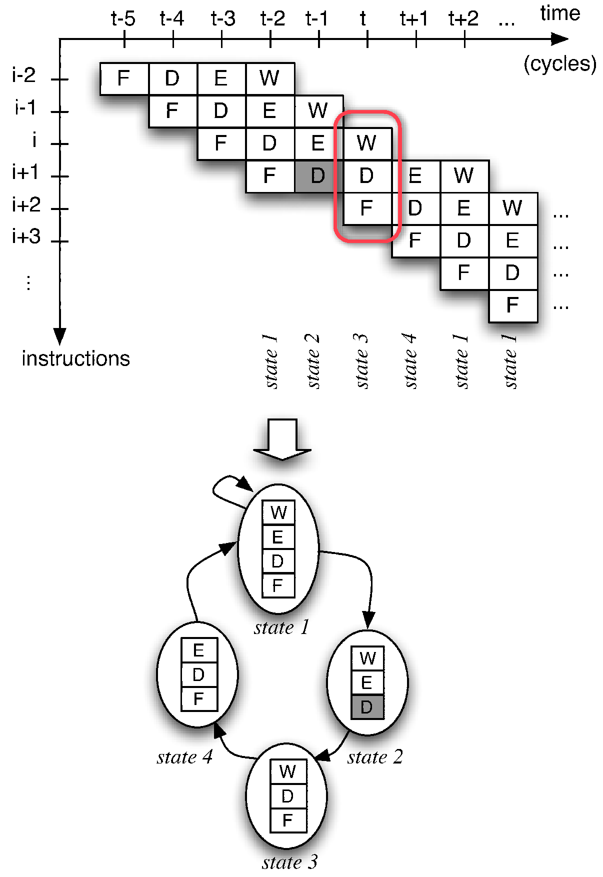


Fig. 2. A state of the automaton represents the state of the pipeline at a given time. In this example with a 4-stages pipeline, three instructions are in the pipeline at time *t*, and the 'D' stage was stalled at time *t-1*. The automaton highlights the pipeline sequence, assuming that there is only one instruction type (this restriction is only made for clarity reason).

### A. Resources

*Resources* are defined as a mechanism to describe temporal constraints in the pipeline. They are used to take into account *structural hazards* and *data hazards* in the pipeline.

Two types of resources are defined, *internal* and *external* resources, that model constraints respectively statically and dynamically.

*1) Internal resources:* can be compared to "resources" in [12]. They model structural hazards.

As the pipeline state is known (*i.e.* the instructions are defined for each stage of the pipeline), then the state of each *internal resource* is fully defined (taken or available). In that case, when the automaton is built (and then the simulator), constraints described by *internal resources* are directly resolved when the set of next states is built.

Internal resources are designed to describe structural hazards inside a pipeline. As these resources are taken into account at build time (*static* approach), no computation overhead is required to check for this type of constraint at runtime.

For example, each pipeline stage is modelled by an internal resource. Each instruction that enters in a stage takes the associated resource, and releases that resource when it leaves. The resulting constraint is that each pipeline stage gets at most one instruction. Another example of internal resource is presented in section VI.

*2) External resources:* represent resources that are *shared* with other hardware components such as timers or memory controllers. It is an extension of internal resources to take into account resources that must be managed *dynamically* (*i.e.* during the simulation). For instance, in the case of a memory controller, the pipeline is locked if it performs a request whereas the controller is busy. Otherwise, the pipeline stage that requests the memory access takes the resource.

If an external resource can be taken by instructions in more than one pipeline stage, a priority is set between stages. For example, at least two pipeline stages may compete for a memory access using a single pipelined micro-controller without instruction cache nor data cache.

One interesting property of the external resources is that it allows to check for data hazards. An external resource is used, associated to a *data dependancy controller*. In this section, we suppose that the first stages of the pipeline are a `fetch` stage, followed by a `decode` stage that reads operands.

The data dependancy controller works as presented in algorithm 1 when instructions in the pipeline are executed. An instruction that is in the `Fetch` stage sends a *request* to the controller to check if all of the operands, that will be taken in the next stage (`Decode`), are available. If at least one of the register is busy (because it is used by one or more instructions in the pipeline), the request fails and the associated external resource is set to `busy`.

When the transition's condition is evaluated to get the next automaton state, the transition associated with this new condition (the transition's condition depends of the state of external resources) will lead to a state that inserts a stall in the pipeline. It allows the instruction that is in the `fetch` stage to wait for its operands, and resolves the data dependancy.

**Algorithm 1**: Instructions and data dependancy controller interaction during simulation.

```
if there is an instruction in the fetch stage and the instruction
will need operands in the decode stage then
    - The instruction sends a request to the controller;
    if at least one register is busy then
        - the request fails: the external resource associated is
        set to busy;
    else
        - the request success: the external resource associated
        is set to available;
else
    - the request success: the external resource associated is set
    to available;
```

### B. Instruction class

To reduce the automaton state space, instructions that use the same resources (internal and external) are grouped to build ***instruction classes***.

The number of instruction classes is limited to $2^{R_{ext}+R_{int}}$ ($R_{ext}$ and $R_{int}$ are the number of respectively external and internal resources in the system), but this maximum is not reached because some resources are shared by all instructions, like pipeline stages, which leads to get a lower number of instruction classes.

## IV. Generating the finite automaton

The automaton represents all the possible simulation scenarios of the pipeline. A state of the automaton represents a state of the pipeline, which is defined as the list of all pair (*instruction classes, pipeline stage*) in the pipeline at a given time. For a system with $c$ *instruction classes*, there are $c + 1$ possible cases for each pipeline stage $s$ of the pipeline (each instruction class or a stall). The automaton is *finite* because it has at most $(c + 1)^s$ states. The initial state is the one that represents an empty pipeline. A transition is taken at each clock cycle and its condition depends *only* on:

- the state of the external resources (taken or available);
- the *instruction class* of the next instruction that can be fetched in the pipeline.

Internal resources are already resolved in the generated automaton and does not appear in the transition's condition. Other instructions in the pipeline are already known for a given automaton state, thus only the next instruction that will be fetched is necessary. This transition's condition is called *basic* condition. As many different conditions can appear to go from one state to another one, the transition's condition is a disjunction of *basic* conditions.

The number of possible transitions is limited to at most $c \times 2^{R_{ext}}$ *for each state* ($c$ is the number of *instruction classes* and $R_{ext}$ is the number of external resources in the system). It implies that there are at most $(c + 2)^s \times 2^{R_{ext}}$ transitions for the whole automaton.

The automaton generator algorithm is presented in algorithm 2 and is based on a breadth-first exploration graph to prevent stack problems. The basic idea of the algorithm is that from the initial state, it computes all the possible basic conditions. From the current state, each possible transition is taken to get the set of next automaton states. The algorithm is then reiterated for each state that have not been processed.

**Algorithm 2**: Generation of the automaton pipeline model.

```
- Create a list that contains the initial automaton state;
- Create an automaton, with the initial automaton state;
while list is not empty do
    - Get an automaton state in the list (start state);
    - Generate all the possible basic conditions (combinations
    of external resources, combined with the instruction class
    of the next instruction fetched);
    for each basic condition do
        - Get the next automaton state (this is a deterministic
        automaton), using the basic condition and the start
        state;
        if the state is not yet included in the automaton then
            - Add the new automaton state (target state) in the
            list;
            - Add the new automaton state in the automaton;
        if the transition does not exist then
            - Create a transition, with an empty condition;
        - Update the transition's condition, by adding a basic
        condition (disjunct);
    - Remove automaton start state from the list;
```

The central function of this algorithm is the one that can *get the next automaton state*, when a basic condition is known. From a generic pipeline model, this function computes the next state of an automaton, taking into account all the constraints brought by resources (internal and external). A pipeline is modelled as an ordered list of pipeline stages, where each pipeline stage is an internal resource. In the algorithm 3, the pipeline stages in the loop are taken from the last to the first, because the pipeline stage that follows the current one must be empty to get a new instruction.

This algorithm allows to detect sink states in the automaton (not shown in the algorithm 3, for clarity reason). A sink state corresponds to a wrong pipeline description.

***Combinatorial explosion*** The increase in complexity of the pipeline to model leads to get a combinatorial explosion. As presented above, the automaton is limited to $(c+1)^s$ states and $(c+2)^s \times 2^{R_{ext}}$ transitions. The maximum size of the automaton increases exponentially with the pipeline depth and the number of external resources, and in a polynomial way with the number of instruction classes. We can discern three types of processors:

- short pipelines (5-6 stages) that can be found in simple processors, generally used in embedded systems (4 stages on the Infineon C167). There is no combinatorial explosion due to the short pipeline;
- processors with a single deep pipeline, called super-pipelines (8 stages with the MIPS R4000). There have both a deep pipeline and many instruction classes. To reduce the complexity of the automaton, These long pipelines can be cut in two parts to genarate two smaller automata that are synchronized using an external resources;
- processors with a pipeline that has many branches: each branch of pipeline may be modelled by a separate automaton as introduced in [12]. Instruction are dispatched among different branches, thus each branch has less instruction classes. This kind of processor can be modelled with several automata. Splitting a complex pipeline into different branches will be studied in future work.

**Algorithm 3**: Function that *gets the next automaton state*, from a given state, with a known basic condition.

```
for each pipeline stage, from the last to the first do
    if there is an instruction class in the current stage then
        if resources required by the instruction class can be
        taken in the next pipeline stage then
            - Instruction class releases resources in the current
            pipeline stage;
            if there is a next pipeline stage then
                - Instruction class is moved in the next
                pipeline stage;
                - Resources required in the next pipeline stage
                are taken;
            - Instruction class is removed from the current
            stage;
```

## V. A brief description of HARMLESS

The HARMLESS ADL allows to describe a hardware architecture using different parts:

- the instruction set;
- the hardware components used by the instructions like memory, registers, ALU, ...;
- the micro-architecture;
- the pipeline;
- the peripherals like timers, input/output, ...

A `component`, in HARMLESS ADL, allows the functional description of a hardware component. It may contain one or many methods. A `method` allows an instruction to access a function offered by a component.

The micro-architecture is described in an `architecture` section. It forms the interface between a set of hardware components and the definition of the pipeline. It allows to express hardware constraints having consequences on the temporal sequence of the simulator. It may contain many `device`s to control the concurrency between instructions to access the same component. Every device in the architecture is related to one component. The different methods of a component can be accessed by a `port` that allows to control the competition during access to one or many methods. A port may be private to the micro-architecture or shared (ie the port is not exclusively used by the micro-architecture). The next section shows two examples that illustrate these notions.

## VI. Experiment

Two examples are presented in this section. The first one is a very simple example which leads to a very small automaton. The second one is a more complex example with a 6 pipeline stages inspired by a DLX simple pipelined architecture [6], using the Freescale XGate instruction set [4].

### A. A simple example

Let's consider an example with a 2 stages pipeline, with only 1 instruction (Nop), 1 component (the Memory) and one temporal constraint (the memory access in the fetch stage). Using the HARMLESS ADL, this pipeline can be described as follow:

```
architecture Generic {
   device mem : Memory {
      shared port fetch : getValue;
   }
}

pipeline pFE  maps to Generic {
   stage F {
       mem : fetch;
   }
   stage E {
   }
}
```

In the description above, two objects are declared: the `architecture` named `Generic` and the `pipeline` named `pFE`.

the architecture contains one device (`mem`) to control the concurrency to access the Memory component. In this description: at a given time, the method `getValue`, that get the instruction code from the memory, can be accessed one time using the `fetch` port. We suppose this access can be made concurrently by other bus masters. So the port is `shared`.

The pipeline `pFE` is mapped to the `Generic` architecture. The 2 stages of pipeline are listed. In stage `F`, an instruction can use the `fetch` port.

A shared port is translated to an external resource `M`. When `M` is available the Memory can be accessed through the `fetch` port. Since there is only one instruction in this example, there is only one instruction class. The instruction class depends on the external resource `M` to enter in stage `F`.

This description leads to generate the 4 states automaton shown in figure 3. For each state, the pipeline state is defined at a given time: 'N' represents an instruction class and '-' represents an empty stage. A transition's condition is composed of an instruction that could be fetched, and the state of the external resource. 'M' and '/M' mean that the external resource is respectively available or busy. An 'X' for the instruction class or an external resource means that the parameter is irrelevant for the transition's condition.

The initial state, on the left, represents an empty pipeline. At the next clock cycle, two transitions may be taken. In both transition's conditions, the instruction class is irrelevant (as there is only one). A transition is taken depending on the state of the external resource. If the resource is busy (transition labelled `X,/M`), the memory controller access is not allowed and no instruction can be fetched, the pipeline remains in the same state. If the external resource is available, an instruction of class 'N' is fetched. The new state of the pipeline is `N-`.

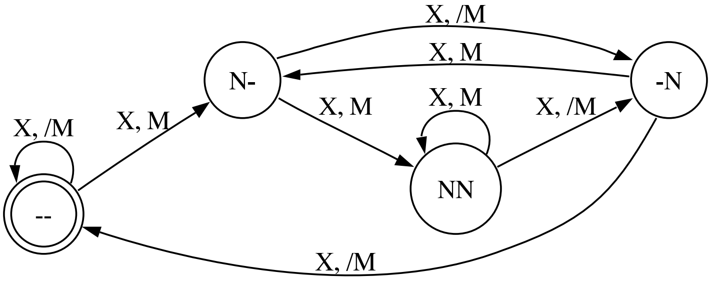


Fig. 3. 4 states automaton generated for the very simple example.

### B. A more realistic example

The second example is more realistic and considers a pipeline with 6 stages. The pipeline is composed of a `Fetch` stage to get the instruction code in memory, a `Decode` stage which decodes the instruction, reads operands and performs branch instructions, 2 `Execute` stages, a `Memory` access stage and a `WriteBack` stage that performs write accesses on the register bank. 88 instructions are available in this example.

The concurrency constraints are:

- the registers file is able to perform 3 reads and 2 writes in parallel;
- an Harvard architecture (separate program and data memories) is used;
- the computation in the ALU requires 2 stages and is not pipelined;

Using the HARMLESS ADL, this 6 stages pipeline can be described as follow:

```
architecture Generic {
   device gpr : GPR {
      port rs1 : read;
      port rs2 : read;
      port rs3 : read;
      port rd1 : write;
      port rd2 : write;
   }
   device alu : ALU {
      port all;
   }
   device mem : Memory {
      shared port fetch : read;
      shared port loadStore : read or write;
   }
   device fetcher : FETCHER {
      port branch : branch;
   }
}
```

```
pipeline pFDEAMWB maps to Generic {
   stage Fetch {
      MEM : fetch;
   }
   stage Decode {
      fetcher : branch;
      gpr : rs1, rs2, rs3, rd1, rd2;
   }
   stage Execute1 {
      alu  release in Execute2 : all;
   }
   stage Execute2 {
   }
   stage Memory {
      mem : loadStore;
   }
   stage WriteBack {
      gpr : rd;
   }
}
```

In the same way, this description declares two objects : `architecture` and `pipeline`. In the `architecture`, many devices are declared. Port `loadStore` allows the access to 2 methods. the keyword 'or' is equivalent to the exclusive or. `loadStore` allows to access the method `read` or `write`. At a given time, if an instruction uses `read` in a stage of pipeline, the second method `write` becomes inaccessible, and the associated resource is set to busy.

Sometimes, using any method of a component makes it unavailable. Instead of forcing the user to give the list of all methods, an empty list is interpreted as a all methods list. Here the `alu` device uses this scheme.

When using a port in a pipeline stage, it is implicitly taken at the start of the stage and released at the end. If a port needs to be held for more than one stage, the stage where it is released is explicitly given. Here port `all` of `alu` is taken in the `Execute1` stage and released in the `Execute2` stage.

As in the previous example, shared ports `loadStore` and `fetch` are translated to external resources. One is associated with the data memory controller. The other one is associated to the program memory controller. A third external resource is used to check for data dependancies during the simulation (see section III-A2).

Other ports may or may not have an associated internal resource. For instance, the `gpr` device offers enough ports to satisfy the needs of the instruction set. So no internal resource is used to constrain the accesses to GPR's methods.

Instruction classes group instructions that use the same resources (internal or external) as presented in section III-B. In this example, 10 resources are used:

- 6 internal resources for the pipeline stages;
- 1 internal resource for the ALU management: `alu`;
- 2 external resources for the memory accesses: `fetch` and `loadStore`;
- 1 external resource to check data dependancies: `dataDep`.

As each instruction depends on the `fetch` resource and the pipeline stages, only three resources can differentiate instructions: there may be a maximum of $2^3 = 8$ instruction classes. We can notice that an architecture without the ALU structural constraint, the maximum of instruction classes would be reduced to $4$. Instructions used in the example (a Fibonacci sequence) are displayed in table I. It uses 5 instruction classes. The 3 remaining instruction classes correspond to impossible configurations.

TABLE I
INSTRUCTIONS USED IN THE EXAMPLE WITH THE XGATE. INSTRUCTIONS THAT USES THE SAME RESOURCES ARE IN THE SAME INSTRUCTION CLASS.

| **opcode** | **Alu** | **loadStore** | **dataDep** | **Inst. class** |
|---|---|---|---|---|
| LDW (load) | | X | X | 1 |
| STW (store) | | X | X | 1 |
| LDH (load) | | X | | 2 |
| LDL (load) | | X | | 2 |
| MOV | | | X | 3 |
| BGT (branch) | | | X | 3 |
| BRA (branch) | | | | 4 |
| ADDL | X | | X | 5 |
| ADD | X | | X | 5 |
| CMP | X | | X | 5 |

This example is executed on an Intel Core 2 Duo @ 2.4 GHz processor with 2 GB RAM. The results are the following: 21.3 s are required to compute the whole simulator. This elapsed time is split in 7.7 s to generate the automaton from the pipeline description, 3.1 s to generate C++ files from the automaton and 10.6 s to compile the C++ files (using GCC 4.0). About 30 MB of RAM are required to generate the automaton. The generated automaton has 43 200 states and 325 180 transitions. 2 030 401 pipeline states were calculated and the generated C++ files represent 4.7 MB in 79 800 lines of code. The simulator generation is fast enough to model realistic processors.

## VII. SIMULATION RESULT

We present in this section the simulation result provided by our tool from the example presented in section VI-B:

- each line represents an instruction that is executed in the pipeline. The instruction that follows a branch instruction (`-`) points out that it is a dummy instruction that is fetched before the branch detection in the decode stage (1). No behaviour is associated with the instruction because the branch instruction is taken;
- each column represents one processor cycle. Numbers from 0 (`Fetch`) to 5 (`Write Back`) represent the pipeline stage number in which the instruction is.

From this short example, we can get both the *functional behaviour* (registers and memory are updated for each instruction) and the *temporal behaviour*. This short example of temporal behaviour shows that:

```
MOV R6,R0         012345
LDL R6,#0x2c       012345
CMP R6,R7           0    12345
BGT 1                    0  12345
-                           012345
MOV R6,R7                    012345
LDW R2,(R5,R6+)               0    12345
LDW R3,(R5,R6+)                    0    12345
ADD R4,R2,R3                            0    12345
STW R4,(R5,R6)                               0    12345
ADDL R7,#0x2                                      012345
BRA -12                                            012345
```

Fig. 4. Execution trace produced by the generated simulator on the Fibonacci sequence

- the four instructions LDW (x2), ADD and STW are data dependent, and only one instruction is executed at each time in the pipeline. No bypass circuitry is included in our description;
- the BGT instruction (line 4) is delayed for 2 cycles because it needs the ALU result from the previous instruction (comparison);

The temporal behaviour is required to compute the WCET of real time applications.

## VIII. From simulation to WCET

The automaton sequencing is directly linked to the processor clock. Thus, the time required to execute an instruction block depends directly on the number of transitions that are taken during the simulation. This property can be integrated in a static WCET approach, for instance using an Implicit Path Enumeration Technique (IPET) approach [11]. In that case, the simulator has in charge to give the execution time of basic blocks on which the IPET algorithm is based to determine the WCET. Additionally, it can give the pipeline state after the execution of the basic block, directly obtained from the last automaton state. Our tool is being integrated with the OTAWA tool [3]: Otawa is a Framework for Experimenting WCET Computations.

On the real example presented in the previous section, it takes 23.9 s to simulate 100 million instructions (requiring 270 million cycles), on an Intel Core 2 duo@2.4GHz. This cycle accurate simulation tool is fast enough to be integrated in such static WCET analysis tools. As a comparison, the Instruction Set Simulator required 6.3 s for the same scenario, but without any temporal information. So the increase factor for computing timing properties is less than 4.

We have focus our study in the pipeline modelling, but other components may significantly influence computation timings, such as caches (branch, instructions or data). Additional delays (that model a cache miss) can be taken into account using external resources, see section III-A2

## IX. Conclusion

This paper has presented the method used in HARMLESS to generate a Cycle Accurate Simulator from the description of a pipeline and its hazards. The method uses an improved version of the BuildForwardFSA algorithm to handle the data and control hazards as well as the concurrent accesses to devices that are not managed statically by the automaton. This improvement is done by using external resources at the cost of a larger automaton. The results look promising. By adding cycle-accurate pipeline simulation to an Instruction Set Simulator, the simulation time is only increased by a factor less than 4. Another important part of this work is the design of an Architecture Description Language HARMLESS [7]. A small part of this language is briefly presented here.. It is a very important part because it considerably simplify the design of a simulator - and so the execution time computation - for a particular target processor. Future work will focus on the minimisation of the automaton, the use of multiple automata to reduce the global size of the tables, and to model and simulate superscalar processors. How to model dynamic superscalar processors, including speculative execution is also planned.

## References

[1] Vasanth Bala and Norman Rubin. Efficient instruction scheduling using finite state automata. In *MICRO 28: Proceedings of the 28th annual international symposium on Microarchitecture*, pages 46–56, Los Alamitos, CA, USA, 1995. IEEE Computer Society Press.

[2] Frank Burns, Albert Koelmans, and Alexandre Yakovlev. Wcet analysis of superscalar processors using simulationwith coloured petri nets. *Real-Time Syst.*, 18(2-3):275–288, 2000.

[3] Hugues Cassé and Pascal Sainrat. Otawa, a framework for experimenting wcet computations. In *European Congress on Embedded Real-Time Software (ERTS)*, page (electronic medium), ftp://ftp.irit.fr/IRIT/TRACES/6278_ERTS06.pdf, janvier 2006. SEE. 8 pages.

[4] Freescale Semiconductor, Inc. *XGATE Block Guide*, 2003.

[5] Ashok Halambi, Peter Grun, and al. Expression: A language for architecture exploration through compiler/simulator retargetability. In *European Conference on Design, Automation and Test (DATE)*, March 1999.

[6] John L. Hennessy and David A. Patterson. *Computer Architecture A Quantitative Approach-Second Edition*. Morgan Kaufmann Publishers, Inc., 2001.

[7] R. Kassem, M. Briday, J.-L. Béchennec, G. Savaton, and Y. Trinquet. Instruction set simulator generation using harmless, a new hardware architecture description language. Unpublished.

[8] Kevin Kranen. *SystemC 2.0.1 User's Guide*. Synopsys, Inc.

[9] J.Y.T. Leung, editor. *Handbook of Scheduling*. Chapman & Hall, CRC Press, 2004.

[10] Xianfeng Li, A. Roychoudhury, T. Mitra, P. Mishra, and Xu Cheng. A retargetable software timing analyzer using architecture description language. In *ASP-DAC '07: Proceedings of the 2007 conference on Asia South Pacific design automation*, pages 396–401, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Yau-Tsun Steven Li and Sharad Malik. Performance analysis of embedded software using implicit path enumeration. In *Workshop on Languages, Compilers, & Tools for Real-Time Systems*, pages 88–98, 1995.

[12] Thomas Müller. Employing finite automata for resource scheduling. In *MICRO 26: Proceedings of the 26th annual international symposium on Microarchitecture*, pages 12–20, Los Alamitos, CA, USA, 1993. IEEE Computer Society Press.

[13] Todd A. Proebsting and Christopher W. Fraser. Detecting pipeline structural hazards quickly. In *POPL '94: Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 280–286, New York, NY, USA, 1994. ACM Press.

[14] Adriano Tavares, Carlos Couto, Carlos A. Silva, and José Lima, C. S. Metrôlho. *WCET Prediction for embedded processors using an ADL*, chapter II, pages 39–50. Springer Verlag, 2005.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 665 – 672

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Software Certification for Safety-Critical Systems: A Status Report

Andrew Kornecki
Dept. of Computer and Software Engineering
Embry-Riddle Aeronautical University
Daytona Beach, FL 32614, USA
kornecka@erau.edu

Janusz Zalewski
Department of Computer Science
Florida Gulf Coast University
Fort Myers, FL 33965-6565, USA
zalewski@fgcu.edu

*Abstract*—**This paper presents an overview and the role of certification in safety-critical computer systems focusing on software and hardware use in the domain of civil aviation. It discusses certification activities according to RTCA DO-178B "Software Considerations in Airborne Systems and Equipment Certification" and RTCA DO-254 "Design Assurance Guidance for Airborne Electronic Hardware." Specifically, certification issues in real-time operating systems, programming languages, software development tools, complex electronic hardware and tool qualification are discussed. Results of an independent industry survey done by the authors are also presented.**

## I. Introduction

CERTIFICATION is the hot issue in many industries that rely on the use of computers and software in embedded systems that control safety-critical equipment. The term "certification" in software engineering is typically associated with three meanings: certifying product, process, or personnel. Product and process certification are the most challenging in developing software for real-time safety critical systems, such as flight control and traffic control, road vehicles, railway interchanges, nuclear facilities, medical equipment and implanted devices, etc. These are systems that operate under strict timing requirements and may cause significant damage or loss of life, if not operating properly. Therefore, the society has to protect itself, and governments and engineering societies initiated establishing standards and guidelines for computer system developers to follow them in designing such systems in several regulated industries, including aerospace, avionics, automotive, medical, nuclear, railways, and others.

Consequently, the U.S. government and international agencies that regulate respective industries have issued a number of standards, guidelines, and reports related to certification and/or other aspects of software assurance, such as licensing, qualification, or validation, in their specific areas of interest. Two such guidance documents for civil aviation, DO-178B [1] and DO-254 [2], developed by RTCA, Inc., describe the conditions for assurance in designing software and electronic hardware in airborne systems. The guidelines are adopted by the U.S. Federal Aviation Administration (FAA) and the European EUROCAE, as mandatory for design and implementation of airborne systems.

In this paper we present an overview of current practices in civil aviation industry and discuss issues related to certification of software and hardware to meet the guidance requirements. Section 2 discusses the role of guidance in certification, and sections 3 and 4 review the certification issues according to DO-178B and DO-254, respectively. Section 5 provides some conclusions.

## II. The Role of Standards in Certification

The RTCA, Inc., previously known as the Radio-Telecommunication Committee for Aviation, is a non-profit corporation formed to advance the art and science of aviation and aviation electronic systems for the benefit of the public. The main RTCA function is to act as a Federal Advisory Committee to develop consensus-based recommendations on aviation issues, which are used as the foundation for Federal Aviation Administration Technical Standard Orders controlling the certification of aviation systems.

In 1980, the RTCA, convened a special committee (SC-145) to establish guidelines for developing airborne systems and equipment. They produced a report, "Software Considerations in Airborne Systems and Equipment Certification," which was subsequently approved by the RTCA Executive Committee and published in January 1982 as the RTCA document DO-178. After gaining further experience in airborne system certification, the RTCA decided to revise the previous document. Another committee (SC-152) drafted DO-178A, which was published in 1985. Due to rapid advances in technology, the RTCA established a new committee (SC-167) in 1989. Its goal was to update, as needed, DO-178A. SC-167 focused on five major areas: (1) Documentation Integration and Production, (2) System Issues, (3) Software Development, (4) Software Verification, and (5) Software Configuration Management and Software Quality Assurance. The resulting document, DO-178B, provides guidelines for these areas [1].

RTCA/EUROCAE DO-254/ED-80 [2] was released in 2000, addressing design assurance for complex electronic hardware. The guidance is applicable to a wide range of hardware devices, ranging from integrated technology hybrid and multi-chip components, to custom programmable micro-coded components, to circuit board assemblies (CBA), to en-

The presented work was supported in part by the Aviation Airworthiness Center of Excellence under contract DTFACT-07-C-00010 sponsored by the FAA. Findings contained herein are not necessarily those of the FAA.

tire line replaceable unit (LRU). This guidance also addresses the issue of COTS components. The document's appendices provide guidance for data to be submitted, including: independence and control data category based on the assigned assurance level, description of the functional failure path analysis (FFPA) method applicable to hardware with Design Assurance Levels (DAL) A and B, and discussion of additional assurance techniques, such as formal methods to support and verify analysis results.

## III. Software Certification according to DO-178B

There are three essential categories of software that impact the certification process, due to their different functionality: real-time operating systems, programming languages (with their compilers), and development tools.

### *A. Real-Time Operating Systems*

There is an evident trend to adopt the Real-Time Operating System (RTOS) kernels to increasing scrutiny of regulatory demands. The vendors have quickly "jumped on the bandwagon" and attempted to comply with requirements of DO-178B, claiming certifiability. This includes VxWorks from Wind River Systems [3-5], as well as LynxOS from LynuxWorks, Integrity from Green Hills Software, Linux and RTLinux, RTEMS and microC.

Romanski reports [3] on certification attempts of VxWorks that started in 1999. At the start of the project, the specifications, documentation, and source code were all analyzed to determine which features need to be removed or changed to support the certification. The analysis showed that the core OS could be certified and many of the support libraries could be included as well, with some restrictions, for example on memory allocation/deallocation functions. The process was largely automated, with a database and CD-ROM materials deliverable to the auditors. Further, Fachet [4] reports on the VxWorks certification process to meet the criteria of IEC 61508, and Parkinson and Kinnan [5] describe the entire development platform for a specific version of the kernel VxWorks 653 to be used in the integrated modular avionics.

Not much information, except articles in trade magazines, is available on other real-time kernels. Applying the definition of certification as "procedure by which a third-party gives written assurance that a product, process or service conforms to specified requirements", Moraes et al. [6] use the risk assessment technique FMEA (Failure Mode and Effect Analysis) to create a metric and analyze data for two kernels RTLinux and RTEMS. The analysis shows that if the threshold to certify the software is set to an estimated risk lower than 2.5%, only RTEMS would be certified.

Interestingly, a well described process of selecting an RTOS according to DO-178B guidelines [7] led to a choice of microC/OS kernel, a relatively unknown although well documented RTOS, available for many years but not much advertised. Verification of this RTOS has been contracted to an independent organization and all requirements-based tests have been completed in 2003.

### *B. Programming Languages*

A similar trend among vendors is visible in the area of programming languages and compilers. In an earlier article, Halang and Zalewski [8] present an overview of programming languages for use in safety-related applications up to 2002, focusing on PEARL, originated and predominantly used in Germany. Their observation with respect to DO-178B and other standards is that " because verification is the main prerequisite to enable the certification of larger software-based solutions, only serious improvements aiming to support the process of program verification will be a step in the right direction."

There are essentially three contenders among languages used in safety-critical systems: Ada, C/C++ and Java, for which DO-178B certifiability is claimed. Due to limited space, we only address Ada and C/C++. The most advanced in this respect seems to be Ada, whose certification attempts go back to the eighties, with roots in compiler validation [9].

*Ada and Compiler Certification.* Santhanam [10] answers the question, what does it mean to qualify a compiler tool suite per DO-178B requirements, and lists the requirements on the object code and the development process, estimating the overwhelming cost of providing evidence. Therefore, defensive techniques are advocated, to assure confidence in the compiler correctness with the use of assertions, optimizations turned off, no suppression of run-time checks, avoidance of nested subprograms, etc.

Features of the object model of Ada 2005 are claimed to be "well suited for applications that have to meet certification at various levels" [11]. It meets the safety requirement, which means that programmers are able "to write programs with high assurance that their execution does not introduce hazards" [12], in order "to allow the system to be certified against safety standards", such as DO-178B. However, the common opinion, expressed by the same authors, who actually developed compilers, is that compilers "are far too complex to be themselves certified" [11]-[12].

One version of Ada, which makes use of its severely limited subset, named SPARK, seems to have gained some popularity in safety-critical applications, because of the existence of its formal definition. Amey et al. [13] report on multiple applications of SPARK in industry, including one to the DO-178B Level A.

*The C/C++ Certification Issues.* In the C/C++ world, there have not been many reports on the successful uses of these languages in safety-critical applications that would pass or be aimed at any certification efforts. The languages are being widely criticized for having features not necessarily suitable for safety-critical systems.

Hatton [14] gave an overview of safer C subsets and MISRA C, in particular, following his crusade towards make C a safer language. His premise was that "C is the perfect language for non-controversial safer subsetting as it is known to suffer from a number of potential fault modes and the fault modes are very well understood in general." He analyzed the standards with respect to style related rules, divided further into rules based on "folklore" and those based on known failures. He observes "MISRA C does not address all known fault modes, and does not incorporate the full range of analysis checks that it might."

Despite the enormous popularity of C++, the number of C++ applications in avionics is relatively low, perhaps due to the multitude of known language problems. Subbiah and Nagaraj [15] report on the issues with C++ certification for avionics systems, focusing on structural coverage, whose intent is "to ensure that all output of the compiler is tested during the execution of the requirement-based tests, so as to preclude the possibility that some instruction or data item produced by the compiler is first depended upon during operation.“

*C.Software Development Tools*

Regarding the use of tools, the FAA recently released a comprehensive report by the current authors on " Assessment of Software Development Tools for Saf ety-Critical Real-Time Systems" [16], which has been summarized in [17] and briefed in [18] regarding tool qualification. The experimental part of this work involved collecting data from the usage of six software design tools (as opposed to verification tools [19]), in a small-scale software development project, regarding four software quality criteria. Assuming these criteria were direct metrics of quality, the following specific measures to evaluate them were defined and used in the experiments:

- usability measured as development effort (in hours)
- functionality measured via the questionnaire (on a 0–5 point scale)
- efficiency measured as code size (in Lines of Code, (LOC))
- traceability measured by manual tracking (in number of defects).
- collection and analysis of results.

Since then, a good number of articles have been written on tool verification, qualification and certification attempts. Regarding software, the tool qualification process must address the requirements of the DO-178B. In particular, the decision must be made, whether tool qualification is necessary (see Fig. 1).

A tool is categorized as the development tool, if it can insert an error in the airborne system, or as the verification tool, if it may only fail to detect an error. In the following, we try to cover issues related to software verification tools.

For verification tool qualification, several interesting papers have been published in the last few years. As Dewar and Brosgol [20] point out in their discussion of static analysis tools for safety certification, a tool as fundamental as the compiler can be certainly treated as a development tool, but also as a verification tool, since compilers "often perform much more extensive tasks of program analysis." As a perfect counterexample they refer to the Spark's Examiner, which is not a usual kind of compiler, because it does not generate code at all. It is only used for checking the program, nevertheless is a part of a software development process. Furthermore, they ask the question should the tools "be certified with the same rigorous approach that is used for safety-critical applications?" Their answer is that this is not practical, and they support this view by stating that even "the compilers themselves are out of reach for formal safety certification, because of their inherent complexity."

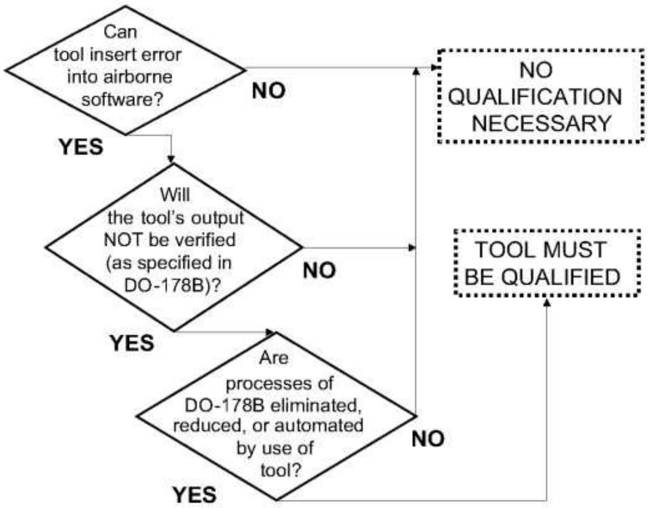

Can tool insert error into airborne software?
NO
YES
NO QUALIFICATION NECESSARY
Will the tool's output NOT be verified (as specified in DO-178B)?
NO
YES
TOOL MUST BE QUALIFIED
Are processes of DO-178B eliminated, reduced, or automated by use of tool?
NO
YES

Fig. 1 Tool qualification conditions according to DO-178B [1].

Dewar supports this view in another article [21], elaborating more on the tools for static analysis of such properties as schedulability, worst-case timing, freedom from race conditions, freedom from side effects, etc. He also offers his views on the use of testing, object-oriented programming, dynamic dispatching, and other issues in developing safety-critical systems. He elaborates on the role of the Designated Engineering Representatives (DER's), whose job is to work with software development companies and the certification authorities on the qualification and certification issues, stating that DER's "are the *building inspectors* of the software engineering industry."

In another article, Santhanam [22] describes a toolset called Test Set Editor (TSE), which automates the compiler testing process and working in combination with the Excel spreadsheet and the homegrown scripts in Tcl/tk significantly contribute to cost savings in constructing structural tests to satisfy FAA certification requirements.

A recent FAA report [19] provides an overview of the verification tools available up to the time of report's publication. One tool not covered in this report, Astrée, is described in [23]. It is a parametric, abstract interpretation based, static analyzer that aims at proving the absence of run-time errors in safety-critical avionics software written in C. The authors, representing Airbus, claim that they succeeded on using the tool on a real-size program "as is", without altering or adjusting it before the analysis. Other issues addressed with this tool, although not described in the paper, include: assessment of worst-case execution time, safe memory use, and precision and stability of floating-point computations. In all that, automatically generated code should be subjected to the same verification and validation techniques as hand-written code.

It may be also worth noting that all established tool vendors have been addressing the DO-178B issues for some time now. One such interesting example is McCabe Software [24]. Their document provides a summary of McCabe IQ tool functionality and explains how the tool can be used to support the DO-178B guidelines. Several other vendors do the same, and the current list of safety-critical software tools can be found on the web [25].

## IV. Certification According to DO-254

### *A. Circuitry Compliance with DO-254*

General Issues . With the progress of microelectronic technologies, the avionics hardware is typically custom generated using programmable logic devices. Field Programmable Logic Arrays (FPGA) and Application Specific Integrated Circuits (ASIC) are two leading implementation technologies. More often the devices include also components containing Intellectual Property (IP) chips with dedicated algorithms or custom made solutions resembling general purpose embedded microprocessor's functionality. All this caused an emergence of RTCA document DO-254 [2], which deals with safety assurance for hardware used in avionics and can be used for other safety-critical applications.

What also contributed to the origins of DO-254 is the fact that avionics companies and designers, facing the rigors of DO-178B requirements, began moving device functionality from software to hardware [26]. As reported by Cole and Beeby in 2004 [27], "There are several schemes that have been used by some to take advantage of a current loophole that allows airborne software functionality to be embedded in firmware or programmable devices. This loophole affectively sidesteps the need to adhere to DO-178B as a software standard." Thus, a new document was introduced that forms the basis for certification of complex electronic hardware, by identifying design lifecycle process, characterizing the objectives, and offering means of complying with certification requirements. The Advisory Circular published subsequently by the FAA [28] clarifies the applicability of DO-254 to custom microcoded components, such as ASIC, PLD, FPGA, and similar. In this section, we discuss recent approaches to hardware certification according to DO-254 covered in the literature.

Miner et al. [29] considered compliance with DO-254, before even the guidance was officially released. In a joint project with the FAA, NASA Langley was developing hardware to gain understanding of the document and to generate an example for training. A core subsystem of the Scalable Processor-Independent Design for Electromagnetic Resilience (SPIDER) was selected for this case study.

Hilderman and Baghai [26] offer an advice to manufacturers to map their existing development processes to those of DO-254. The strategy they recommend is "to focus on ensuring correctness at the conceptual design stage and then preserve the design integrity" as one proceeds through the development stages. Each individual vendor or designer faces multiple specific design problems that must be addressed to meet the DO-254 requirements. How they proceed depends on the vendor and the type of problem.

In the white paper of the DO-254 Users Group [30], Baghai and Burgaud offer a package including the following items designed to assist in the qualification process:

- The processes documents, that help define, benchmark and improve the industrial design, verification, validation, and quality assurance processes
- The quality assurance checklists, for reviews and audits, ensuring that each project is compliant with the defined industrial process
- The tools for requirements management and traceability, checking compliance of HDL code with coding standards, HDL code verification, and test suite optimization
- The tools integration into the industrial process, until their qualification (interfaces, report generation for a certification audit, trainings, tools assessment, etc.), and the DO-254 TRAINING by consulting partners.

Cole and Beeby [27] studied DO-254 compliance for graphic processors, considered common off-the-shelf components (COTS), and proposed a multiphase approach to meet DO-254 requirements:

- Provision of a DO-254 COTS data pack to support the use of a given electronic part.
- Provision of a DO-254 compliance statement.
- Process improvement and further analysis.
- Ongoing support for new parts and processes.

Glazebook [31] discussed certification according to DO-254 in the British context, especially the 26 data items listed in the standard as the compliance suite, of which four are required for submission: (a) Plan for Hardware Aspects of Certification; (b) Hardware Verification Plan; (c) Top Level Drawings; and (d) Hardware Accomplishment Summary. He made eight recommendations summarized in the paper.

Barco-Siles S.A. [32] report on the way the company deals with increasing demands related to implementing DO254 causing non-negligible cost, but bringing some advantages. The guidance obliges the supplier to analyze in detail processes, methodologies and tools and to apply a rigorous quality assurance. It also allows the supplier to adapt its set of internal processes to the design assurance level targeted, to optimize efforts while requiring the subcontractor to respect a structured development processes. The resulting products have improved quality and the development cycles are optimized. Verification is focused on design errors, and effort and resources are better distributed. Applying the DO254 gives the assurance that the applicant can obtain from the subcontractor a good level of quality, good documentation, and the ability to reuse the design, if necessary.

When the complexity of designs increases, it is more and more difficult to verify the correctness of circuits and thus their compliance with the specifications. As Karlsson and Forsberg point out [33], "…tests and deterministic analysis must demonstrate correct operation under all combinations and permutations of conditions down to the gate level of the device." To comply with the requirements of DO-254 they developed a design strategy that relies on a semi-formal solution, a hybrid of static and dynamic assertion based verification. They believe that by such independent assessment using their method of tool outputs, the tool qualification will become unnecessary.

EDA Industry Views . Chip and board manufacturers are eager to comply with DO-254, due to their concerns about the market share. Since compliance with the guidance is considered a technological advantage, most of the vendors began changing their development processes towards meeting the DO-254 criteria. Several companies announced their readiness to comply with certification requirements.

Mentor Graphics is particularly aggressive in providing compliance of their products with DO-254. Lange and Boer [34] give an overview of functional hardware verification methodologies, as a part of the design process. They observe that the verification techniques that served well the designs 10-15 years ago are no longer adequate due to a tremendous increase in design complexity and integration. As a consequence, design verification has become a limiting factor in safety-critical systems, with respect to such issues as: complexity, concurrency and metastability. Latest verification techniques are described that handle problems such as state explosion, design traceability and the effectiveness of coverage.

Advanced Verification Methodology (AVM), consisting of constraint random test generation, a total coverage model, design intent specification, and formal model checking, described in [35], has been used on a practical design of FPGA based DMA engine at Rockwell-Collins. The approach based on an open source Transaction Level Modeling (TLM) class library, is vendor neutral and supports SystemVerilog and SystemC standard languages. Due to the open source nature AVM allows code inspections that may be required for certification. Although the project has not been fully completed at the time of this writing, it is believed that AVM helps not only demonstrate that the DO-254 guidelines are followed, but also assists in shortening design cycles.

These verification steps/techniques must be performed in concert with the RTL design, ultimately leading to automatic circuit synthesis [36]. Since automatic synthesis and conversion to gate-level designs is often done with optimizations by the hardware design tools, it may be counterproductive in safety-critical designs, which mandate strict adherence to the guidance. DO-254 defines tool qualification, "to ensure that tools used to design and verify hardware perform to an acceptable level of confidence on the target project." The paper comments on three methods of DO-254 allowed tool assessment: relevant history, independent output assessment, and tool qualification. Since proving relevant history and qualifying the tool are both tedious and expensive processes, requiring the submittal of data, which may not be easily available, the paper suggests the product assessment route to demonstrate that "the hardware item must be thoroughly verified against the functional requirements", thus, the independent tool assessment is not necessary. In the opinion of current authors, the tool output is still an abstract entity, not the hardware item yet, and may contain errors that cannot be detected during verification.

Lee and Dewey [37] shed more light on meeting DO-254 guidance in a form acceptable to the DER, by explicitly proposing:

- requirements management and tracking, with the use of such tools as Reqtify or DOORS
- Register Transfer Level (RTL) code validation, with an automated method to measure RTL to a company standard
- verification process assurance, with the use of AVM, and
- producing design documentation, from requirements, to the RTL code, to the bit streams or Graphic Data System (GDS) II file format.

The mindset of the paper is that "DO-254 is not a burden but a set of guides that helps standardize hardware systems assurance, making flight systems safe."

Aldec and Actel, working in alliance, published some information on their efforts towards making their products DO-254 certifiable. Sysenko and Pragasam [38] outlined their process for airborne systems design assurance which relies on the verification methodology called Hardware Embedded Simulation (HES) and follows two traditional steps: RTL simulation and gate-level simulation. It is a hardware-software simulation platform driven by software that facilitates the implementation of the design in a reconfigurable hardware, such as an FPGA, and then verification of the design functions. Earlier, Land and Bryant [39] presented more details on the process, with MIL-STD-1553 bus chip design as an example to comply with DO-254.

Lundquist in his thesis [40] looked at the problems that arise when trying to DO-254 certify system-on-chip solutions. Since more than 700 Actel FPGAs are used in the Airbus A380 commercial airliner, the Actel Fusion FPGA chip with integrated analog and digital functionality was tested according to the verification guidance. The results have shown that a certification procedure for a standard non-embedded FPGA based safety critical system is possible. However, the question of how these embedded chips could pass certification to be used in safety-critical systems has not been answered.

### B. Tool Certification against DO-254

Since the growing complexity of electronics hardware requires the use of automatic software tools, the DO-254 document also includes a section on tool qualification. It distinguishes between design tools, which can introduce errors into the product, and verification tools, which do not introduce errors into the product but may fail detecting errors in the product. The qualification process tool vendors have to comply with is shown in Figure 2.

Several vendors recently began dealing with tool qualification. Aldec [41] used a sample design of a system containing two connected boards: Aldec HES board (HES-3X3000EX) generating stimuli and collecting results for Design Under Test (DUT) and the second user designed board. The verification process contained three independent stages: simulation, verification, and comparison.

Lange [42] addresses circuit metastability in the context of DO-254 tool certification. Metastability describes what happens in digital circuits when the clock and data inputs of a flip-flop change values at approximately the same time. This

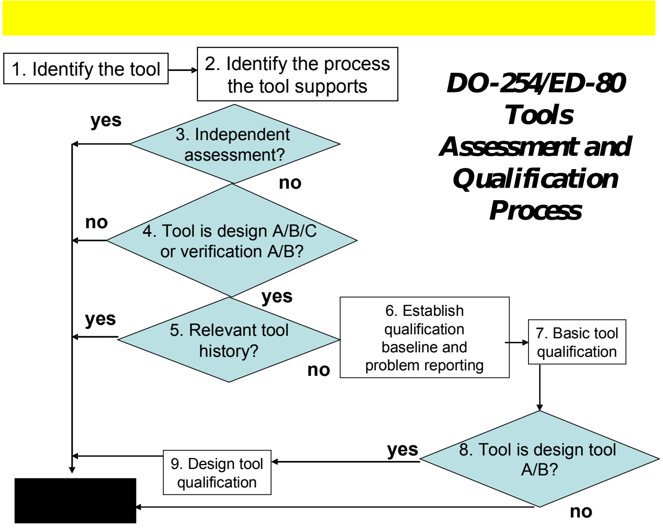

1. Identify the tool
2. Identify the process the tool supports
DO-254/ED-80 Tools Assessment and Qualification Process
yes
3. Independent assessment?
no
no
4. Tool is design A/B/C or verification A/B?
yes
yes
5. Relevant tool history?
no
6. Establish qualification baseline and problem reporting
7. Basic tool qualification
yes
8. Tool is design tool A/B?
9. Design tool qualification
no

Fig. 2 Tool assessment and qualification process according to DO-254 [2].

leads to the flip-flop output oscillating and not settling to a value within the appropriate delay window. This happens in designs containing multiple asynchronous clocks, when two or more discrete systems communicate. Metastability is a serious problem in safety-critical designs as it causes intermittent failures. A comprehensive verification solution is offered by Mentor Graphics 0-In Clock Domain Crossing (CDC) tool. The tool provides an added assurance that the design will function correctly within the intended system. If one has a specific requirement from the customer or a DER to verify the clock domain crossings and identify and eliminate instances of metastability, then one has to use one of the tool assessment methods. Again, the one suggested is the Independent Output Assessment.

Another verification tool from Mentor Graphics, ModelSim, is discussed by Lange [43] in a view of meeting the DO-254 guidance. The paper outlines the exact ten steps to go through the DO-254 assessment and qualification process, as presented in Figure 2. The suggested way to proceed with tool assessment is to avoid qualification by using an independent output assessment method (Step 3 in Figure 2).

TNI [44] presents Reqtify, tool supporting requirement traceability, impact analysis and automated documentation generation which a ccording to DO-254 classification is a verification tool. Prior to the use of the tool, a tool assessment should be performed to ensure that the tool is capable of performing the particular verification activity to an acceptable level of confidence. The assessment is limited to those functions of the tool used for a specific hardware life cycle activity, not the entire tool.

Dellacherie et al. [45] describe a static formal approach that could be used, in combination with requirements traceability features, to apply formal methods in the design and verification of hardware controllers to support such protocols as ARINC 429, ARINC 629, MIL-STD-1553B, etc. A tool name imPROVE-HDL, a formal property checker, has been used in the design and verification of airborne electronic hardware. Reqtify tool has been used to track the requirements throughout the verification process and to produce coverage reports. According to the authors, using imPROVE-HDL coupled with Reqtify gives confidence that the designers can assure that their bus controllers meet the guidelines outlined in DO-254.

### *C.Tool Questionnaire*

To identify issues and concerns in tool qualification and certification, and help understand the underlying problems, we conducted a survey to collect data on the experiences and opinions concerning the use of programmable logic tools as applied to design and verification of complex electronic hardware according to the RTCA DO-254 guidelines. The objective was to collect feedback, from industry and certification authorities, on assessment and qualification of these tools.

The questionnaire has been developed and distributed during the 2007 National FAA Software & Complex Electronic Hardware Conference, in New Orleans, Louisiana, in July 2007, attended by over 200 participants. In subsequent months, we have also distributed this questionnaire to the participants of two other professional events. It has been made available via DO-254 Users Group website ( `http://www.do-254.org/?p=tools` ). As a result of these activities a sample of almost forty completely filed responses was received. Even though this may not be a sample fully statistically valid, the collected results make for several interesting observations.

The survey population, by type of the organization, included the majority of respondents from avionics or engine control developers (65%). Over 95% of respondents have technical background (55% bachelor and 45% master degrees) and over 72% have educational background in electronics. While 97% of respondents have more than three years of experience, 59% have more than 12 years. The most frequent respondents' roles relevant to the complex electronics tools include:

- use of the tools for development or verification of systems (62%)
- managing and acting as company's designated engineering representative (26%)
- development of the tools (2%)
- development of components (12%).

The respondents' primary interest was divided between verification (32%), development (27%), hardware (22%) and concept/architecture (18%).

Considering criteria for the selection of tools for use in DO-254 projects (Figure 3), as the most important have been reported the following: the available documentation, ease of qualification, previous tool use, and host platform, followed by the quality of support, tool functionality, tool vendor reputation, and the previous use on airborne project. Selection of a tool for the project is based either on a limited familiarization with the demo version (50%) or an extensive review and test (40%). The approach to review and test the tool by training the personnel and using trial period on a smaller project seems to be prevailing.

For those who have experienced effort to qualify programmable logic tools (only 14% of respondents), the quality of the guidelines is sufficient or appropriate (62%), so is the ease of finding required information (67%), while the increase of workload was deemed negligible or moderate (80%). An interesting observation concerns the scale of safety improvement due to qualification: marginal (43%), moderate (21%), noticeable (7%) and significant (29%). Similarly, the question about errors found in the tools may be a source for concern: no errors (11%), few and minor errors (50%), significant and numerous (17%). Despite all this, the satisfaction level towards programmable logic tools was high: more than 96% of respondents marked their satisfaction level as 4 out of 5.

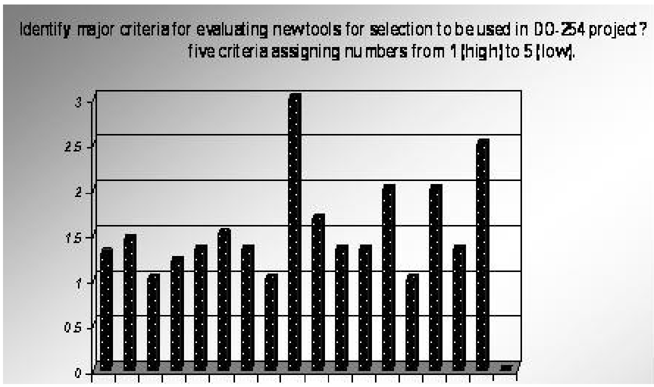


Fig. 3 Tool selection criteria in DO-254 projects (from left to right: vendor reputation, functionality, acquisition cost, compatibility with existing tools, compatibility with development platform, reliability, availability of training, amount of training needed, documentation quality, quality of support, previous familiarity with the tool, performance on internal evaluations, host platform, compatibility with PLDs, previous use on airborne products, tool performance, ease of qualification, other criteria).

Overall, it is obvious that software tools used in design and verification of complex electronics in safety-critical applications should be scrutinized because of concerns that they may introduce design errors leading to accidents. However, the conducted survey indicated that the most important criteria for tool selection are considered to be: available documentation, ease of qualification, and previous tool use, none of which is technical. In this view, work should be done on developing more objective criteria for tool qualification and conducting experiments with tools to identify their most vulnerable functions that may be a source of subsequent design faults and operational errors. Some of the authors specifically point out that the lack of research investment in certification technologies will have a significant impact on levels of autonomous control approaches that can be properly flight certified, and could lead to limiting capability for future autonomous systems.

## V. Summary and Conclusion

The paper makes an attempt to show the role of software certification in development of dependable systems, both from the software and hardware perspective. An important observation is about the increasing role of software tools, which are used to create and verify both software and hardware. An extensive literature review has been presented, focusing on the issues of civil aviation guidance requiring specified level of assurance for the airborne systems, both from the software and hardware perspective.

Both DO-178B and DO-254 guidelines serve industry well and promote rigor and scrutiny required by highly critical systems. However, the relative vagueness of these guidelines causes significant differences in interpretation by industry and should be eliminated. RTCA called a new committee, SC-205, with a charge to revise DO-178B guidance. Possibly, a common ground should be found between RTCA DO-254 and DO-178B guidelines.

## References

[1] RTCA DO-178B (EUROCAE ED-12B), *Software Considerations in Airborne Systems and Equipment Certification*, RTCA Inc., Washington, DC, December 1992.

[2] RTCA DO-254 (EUROCAE ED-80), *Design Assurance Guidance for Airborne Electronic Hardware* , RTCA Inc., Washington, DC, April 2000.

[3] Romanski G., Certification of an Operating System as a Reusable Component, *Proc. DASC'02, 21st Digital Avionics Systems Conf.*, Irvine, Calif., October 27-21, 2002, pp. 5.D.3-1/9.

[4] Fachet R., Re-use of Software Components in the IEC-61508 Certification Process, *Proc. IEE COTS & SOUP Seminar*, London, October 21, 2004, pp. 8/1-17.

[5] Parkinson P., L. Kinnan, *Safety-Critical Software Development for Integrated Modular Avionics*, White Paper, Wind River Systems, Alameda, Calif., November 2007.

[6] Moraes R. et al., Component-Based Software Certification Based on Experimental Risk Assessment, *Proc. LADC 2007, 3rd Latin-American Symposium on Dependable Computing*, Morelia, Mexico, September 26-28, 2007, pp. 179-197.

[7] Maxey B., COTS Integration in Safety Critical Systems Using RTCA/DO-178B Guidelines, Proc. ICCBSS 2003, 2nd International Conference on COTS-Based Software Systems, Ottawa, Ont., February 10-13, 2003, pp. 134-142.

[8] Halang W., J. Zalewski, Programming Languages for Use In Safety Related Applications, *Annual Reviews in Control*, Vol. 27, pp. 39-45, 2003.

[9] Goodenough J.B., The Ada Compiler Validation Capability, *ACM SIGPLAN Notices* , Vol. 15 , No. 11, pp. 1-8, November 1980.

[10] Santhanam V., The Anatomy of an FAA-Qualifiable Ada Subset Compiler, *Ada Letters*, Vol. 23, No. 1, March 2003, pp. 40-43 (Proc. SIGAda'02, Houston, Texas, December 8-12, 2002).

[11] Comar C., R. Dewar, G. Dismukes, Certification & Object Orientation: The New Ada Answer, *Proc. ERTS 2006, 3rd Embedded Real-Time Systems Conference* , Toulouse, France, January 25-27, 2006.

[12] Brosgol B.M., Ada 2005: A Language for High-Integrity Applications, *CrossTalk – The Journal of Defense Systems* , Vol. 19, No. 8, pp. 8-11, August 2006.

[13] Amey P., R. Chapman, N. White, Smart Certification of Mixed Criticality Systems, *Proc. Ada-Europe 2005, 10th Intern. Conf. on Reliable Software Technologies*, York, UK, June 20-24, 2005, pp. 144-155.

[14] Hatton L., Safer Language Subsets: An Overview and Case History - MISRA C, *Information and Software Technology*, Vol. 46, No. 7, pp. 465-472, 2004.

[15] Subbiah S., S. Nagaraj, Issues with Object Orientation in Verifying Safety-Critical Systems, *Proc. ISORC'03, 6th International IEEE Symposium on Object-Oriented Real-Time Distributed Computing* , Hakodate, Hokkaido, Japan, May 14-16, 2003.

[16] Kornecki A., N. Brixius, J. Zalewski, *Assessment of Software Development Tools for Safety-Critical Real-Time Systems*, Technical Report DOT/FAA/AR-06/36, Federal Aviation Administration, Washington, DC, July 2007.

[17] Kornecki A., J. Zalewski, Experimental Evaluation of Software Development Tools for Safety-Critical Real-Time Systems, *Innovations in Systems and Software Engineering – A NASA Journal*, Vol. 1, No. 2, pp. 176-188, September 2005.

[18] Kornecki A., J. Zalewski, The Qualification of Software Development Tools from the DO-178B Certification Perspective, *Crosstalk - The Journal of Defense Software Engineering*, Vol. 19 , No. 4, pp. 19-23, April 2006.

[19] Santhanam V. et al, *Software Verification Tools Assessment Study*, Technical Report DOT/FAA/AR-06/54, Federal Aviation Administration, Washington, DC, June 2007.

[20] Dewar R., B. Brosgol, Using Static Analysis Tools for Safety Certification, *VMEbus Systems* , pp. 28-30, April 2006.

[21] Dewar R.B.K., Safety Critical Design for Secure Systems, *EE Times-India*, July 2006.

[22] Santhanam U., Automating Software Module Testing for FAA Certification, *Ada Letters*, Vol. 21, No. 4, pp. 31-37, December 2001 (Proc. SIGAda'01, Bloomington, MN, Sept. 30 – Oct. 4, 2001).

[23] Souyris J., D. Delmas, Exterimental Assessment of Astreé on Safety-Critical Avionics Software, *Proc. SAFECOMP 2007, 26th Intern. Conf. on Computer Safety, Reliability and Security*, Nuremberg, Germany, Sept. 18-21, 2007.

[24] *DO-178B and McCabe IQ*, McCabe Software, Warwick, RI, December 2006.

[25] *Safety Critical Systems Club Tools Directory*, London, UK, http://www.scsc.org.uk/tools.html

[26] Hilderman V., T. Baghai, Avionics Hardware Must Now Meet Same FAA Requirements as Airborne Software, *COTS Journal*, Vol. 5, No. 9, pp. 32-36, September 2003.

[27] Cole P., M. Beeby, Safe COTS Graphics Solutions: Impact of DO-254 on the Use of COTS Graphics Devices for Avionics, *Proc. DASC'04, 23rd Digital Avionics Systems Conference*, Salt Lake City, Utah, October 24-28, 2004, pp. 8A2-8.1/7.

[28] Federal Aviation Administration, *Advisory Circular AC 20-152, RTCA Document RTCA/DO-254 Design Assurance Guidance for Airborne Electronic Hardware*, June 30, 2005.

[29] Miner P.S. et al., A Case-Study Application of RTCA DO-254: Design Assurance Guidance for Airborne Electronic Hardware, *Proc. DASC 2000, 19th Digital Avionics Systems Conference*, Philadelphia, PA, October 7-13, 2000, Vol. 1, pp. 1A1/1 – 1A1/8.

[30] Baghai T., L. Burgaud, *DO254 Package Process and Checklists: Overview & Compliance with RTCA/DO-254 Document*, White Paper, DO-254 Users Group, March 2004.

[31] Glazebrook I., *The Certification of Complex Hardware Programmable Logic Devices (PLDs) for Military Applications*, White Paper, DNV UK, London, 2007.

[32] Pampagnin P., J.F. Menis, *DO254-ED80 for High Performance and High Reliable Electronic Components*, Internal Paper, Barco-Siles S.A., Peynier, France, 2007.

[33] Karlsson K., H. Forsberg, Emerging Verification Methods for Complex Hardware in Avionics, *Proc. DASC '05, 24th Digital Avionics Systems Conference*, Washington, DC, Oct.-30-Nov. 3, 2005, Vol. 1, pp. 6.B.1-1/11.

[34] Lange M., T.J. Boer, *Effective Functional Verification Methodologies for DO-254 Level A/B and Other Safety-Critical Devices*, White Paper, Rev. 1.1, Mentor Graphics, Wilsonville, Ore., 2007.

[35] Keithan J.P. et al., The Use of Advanced Verification Methods to Address DO-254 Design Assurance, *Proc. 2008 IEEE Aerospace Conference*, Big Sky, Montana, March 1-8, 2008.

[36] Lange M., T. Dewey, Achieving Quality and Traceability in FPGA/ASIC Flow for DO-254 Aviation Projects, *Proc. 2008 IEEE Aerospace Conference* , Big Sky, Montana, March 1-8, 2008.

[37] Lee M., T. Dewey, Accelerating DO-254 for ASIC/FPGA Designs, *VME and Critical Systems*, pp. 28-30, June 2007.

[38] Sysenko I., R. Pragasam, Hardware-based Solution Aides: Design Assurance for Airborne Systems, *Military Embedded Systems*, pp. 26-28, July 2007.

[39] Land I., I. Bryant, FPGA IP Verification for Use in Severe Environments, *Proc. 2005 Annual MAPLD International Conference*, Washington, DC, Sept. 7-9, 2005.

[40] Lundquist P., *Certification of Actel Fusion according to RTCA DO-254*. Master Thesis, Report LiTH-ISY-EX-ET-07/0332-SE, Linköping University, Sweden, May 4, 2007.

[41] Aldec Corp., *DO-254 Hardware Verification: Prototyping with Vectors Mode*. White Paper, Rev. 1.2, Henderson, Nevada, June 2007.

[42] Lange M., Automated CDC Verification Protects Complex Electronic Hardware from Metastability Issues, *VME Critical Systems*, Vol. 26, No. 3, pp. 24-26, August 2008.

[43] Lange M., *Assessing the ModelSim Tool for Use in DO-254 and ED-80 Projects*, White Paper, Mentor Graphics Corp., Wilsonville, Ore., May 2007.

[44] Baghai T., L. Burgaud, *Reqtify: Product Compliance with RTCA/DO-254 Document*, TNI-Valiosys, Caen, France, May 2006

[45] Dellacherie S., L. Burgaud, P. di Crescenzo, Improve – HDL: A DO-254 Formal Property Checker Used for Design and Verification of Avionics Protocol Controllers, *Proc. DASC'03, 22nd Digital Avionics Systems Conf.*, Indianapolis, Oct. 12-16, 2003, Vol. 1, pp. 1.A.1-1.1-8.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Modeling Real-Time Database Concurrency Control Protocol Two-Phase-Locking in Uppaal

Martin Kot
Center for Applied Cybernetics
Dept. of Computer Science,
Technical University of Ostrava
17. listopadu 15,708 33 Ostrava – Poruba
Czech Republic
Email: martin.kot@vsb.cz

***Abstract*—Real-time database management systems (RT-DBMS) are recently subject of an intensive research. Model checking algorithms and verification tools are of great concern as well. In this paper we show some possibilities of using a verification tool Uppaal on some variants of pessimistic concurrency control protocols used in real-time database management systems. We present some possible models of such protocols expressed as nets of timed automata, which are a modeling language of Uppaal.**

## I. Introduction

MANY real-time applications need to store some data in a database. It is possible to use traditional database management systems (DBMS). But they are not able to guarantee any bounds on a response time. This is the reason why so-called real-time database management systems (RTDBMS) emerged.

Research in RTDBMS focused on evolution of transaction processing algorithms, priority assignment strategies and concurrency control techniques. But the research was based especially on simulation studies. Hence at Technical university of Ostrava, Václav Król, Jindřich Černohorský and Jan Pokorný designed and implemented an experimental real-time database system called V4DB [6], which is suitable for study of real time transaction processing. The system is still in further development but some important results were obtained already.

Formal verification is of great interest recently and finds its way quickly from theoretical papers into a real live. It can prove that a system (or more exactly a model of a system) has a desired behavior. The difference between testing and formal verification is that during testing only some possible computations are chosen. Formal verification can prove correctness of all possible computations. A drawback of formal verification is that for models with high descriptive power are almost all problems undecidable. It is important to find a model with an appropriate descriptive power to capture a behavior of a system, yet with algorithmically decidable verification problems.

In this paper we consider so called model checking (see e.g. [3], [8]). This form of verification uses a model of a system in some formalism and a property expressed usually in the form of formula in some temporal logic. Model checking algorithm checks whether the property holds for the model of a system. There are quite many automated verification tools which implement model checking algorithms. Those tools use different modeling languages or formalisms and different logics.

The idea of the research described in this paper came from authors of V4DB. They were interested in using a verification tool on their system. They would like to verify and compare different variants of algorithms and protocols used in RTDBMS. To our best knowledge, there are only rare attempts of automated formal verification of real-time database system. In fact we know about one paper ([9]) only where authors suggested a new pessimistic protocol and verified it using Uppaal. They presented two small models covering only their protocol.

There is not any verification tool intended directly for real-time database systems. We have chosen the tool Uppaal because it is designed for real-time systems. But, it is supposed to be used on so-called reactive systems, which are quite different from database systems. So we need to solve the problem of modeling data records of the database and some other problems. Then we would like to check some important properties of used protocols and algorithms, for example: absence of a deadlock when using an algorithm which should avoid deadlock in the transaction processing, processing transaction with bigger priority instead of transactions with smaller priority and so on.

Big problem of verification tools is so called state space explosion. Uppaal is not able to manage too detailed models. On the other hand, too simple models can not catch important properties of a real system. So we need to find a suitable level of abstraction.

One of the most important and crucial parts of RTDBMS is concurrency control. There were many different concurrency control protocols suggested. In this paper, we will concentrate on variants of a pessimistic protocol called two-phase-locking (2PL). First variant is basic 2PL protocol, then slightly modified version where deadlines are used to abort waiting transactions and finally 2PL – high priority where a

Author acknowledges the support by the Czech Ministry of Education, Grant No. 1M0567.

transaction with higher priority can restart a transaction with a smaller priority. We will show that it is possible to model those protocols, to some level of abstraction, using modeling language of Uppaal. These examples will show possibilities of modeling other similar pessimistic protocols and even some other parts of RTDBMS. The models are inspired by the RTDBMS system V4DB in some way. V4DB is experimental so it has some simplifications which we can use to obtain simpler models yet with important behavior covered (as V4DB has). But it is possible to use ideas shown in this paper for verification of concurrency control algorithms in general.

We will also mention a few simple formula with an Uppaal's answer to show model checking possibilities on suggested models.

## II. Verification tool Uppaal

Uppaal ([2], [4]) is a verification tool for real-time systems. It is jointly developed by Uppsala University and Aalborg University. It is designed to verify systems that can be modeled as networks of timed automata extended with some further features such as integer variables, structured data types, user defined functions, channel synchronization and so on.

A timed automaton is a finite-state automaton extended with clock variables. A dense-time model, where clock variables have real number values and all clocks progress synchronously, is used. In Uppaal, several such automata working in parallel form a network of timed automata.

An automaton has locations and edges. Each location has an optional name and invariant. An invariant is a conjunction of side-effect free expressions of the form $x < e$ or $x \leq e$ where $x$ is a clock variable and $e$ evaluates to an integer. Each automaton has exactly one initial location.

Particular automata in the network synchronize using channels and values can be passed between them using shared variables. A state of the system is defined by the locations of all automata and the values of clocks and discrete variables. The state can be changed in two ways - passing of time (increasing values of all clocks) and firing an edge of some automaton (possibly synchronizing with another automaton or other automata).

Some locations may be marked as committed. If at least one automaton is in a committed location, time passing is not possible, and the next change of the state must involve an outgoing edge of at least one of the committed locations.

Each edge may have a select, a guard, a synchronization and an assignment. Select gives a possibility to choose non-deterministically a value from some range. Guard is a side-effect free expression that evaluates to a boolean. The guard must be satisfied when the edge is fired. Synchronization label is of the form $Expr!$ or $Expr?$ where $Expr$ evaluates to a channel. An edge with $c!$ synchronizes with another edge (of another automaton in the network) with label $c?$. Both edges have to satisfy all firing conditions before synchronization. There are urgent channels as well – synchronisation through such a channel have to be done in the same time instant when it is enabled (it means, time passing is not allowed if a synchronisation through urgent channel is enabled). An assignment is a comma separated list of expressions with a side-effect. It is used to reset clocks and set values of variables.

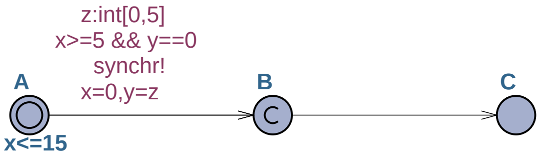


Fig. 1. Graphical representation of a timed automaton in Uppaal

Figure 1 shows how the described notions are represented graphically in Uppaal. There are 3 locations named `A`, `B` and `C`. Location `A` is initial and `B` is committed. Moreover `A` has an invariant `x<=15` with the meaning that the automaton could be in this location only when the value of the clock variable `x` is less or equal 15. The edge between `A` and `B` has the select `z:int[0,5]` – it nondeterministically chooses an integer value from the range 0 to 5 and stores it in variable `z`. This edge also has the guard `x>=5 && y==0`. This means that it can be fired only when the value of the clock variable `x` is greater or equal 5 and the integer variable `y` has the value 0. Data types of variables are defined in a declaration section. Further it has synchronization label `synchr!` and an assignment `x=0, y=z` reseting the clock variable `x` and setting the value of `z` to the integer variable `y`.

Uppaal has some other useful features. Templates are automata with parameters. These parameters are substituted with given arguments in the process declaration. This enables easy construction of several alike automata. Moreover, we can use bounded integer variables (with defined minimal and maximal value), arrays and user defined functions. These are defined in declaration sections. There is one global declaration section where channels, constants, user data types etc. are specified. Each automaton template has own declaration section, where local clocks, variables and functions are specified. And finally, there is a system declaration section, where global variables are declared and automata are created using templates.

Uppaal's query language for requirement specification is based on CTL (Computational Tree Logic, [5]). It consist of path formulae and state formulae. State formulae describe individual states and path formulae quantify over paths or traces of the model.

A state formula is an expression that can be evaluated for a state without looking at the behavior of the model. For example it could be a simple comparison of a variable with a constant `x <= 5`. The syntax of state formulae is similar to the syntax of guards. The only difference is that in a state formula disjunction may be used.

There is a special state formula `deadlock`. It is satisfied in all deadlock states. The state is deadlock if there is not any action transition from the state neither from any of its delay successors.

Path formulae can be classified into *reachability*, *safety* and *liveness*. Reachability formulae ask if a given state formula is satisfied by some reachable state. In Uppaal we use syntax `E<>` $\varphi$ where $\varphi$ is a state formula.

Safety properties are usually of the form: “something bad will never happen”. In Uppaal they are defined positively: “something good is always true”. We use `A[]` $\varphi$ to express, that a state formula $\varphi$ should be true in all reachable states, and `E[]` $\varphi$ to say, that there should exist a maximal path such that $\varphi$ is always true.

There are two types of liveness properties. Simpler is of the form: “something will eventually happen”. We use `A<>` $\varphi$ meaning that a state formula $\varphi$ is eventually satisfied. The other form is: “leads to a response”. The syntax is $\varphi$ `-->` $\psi$ with the meaning that whenever $\varphi$ is satisfied, then eventually $\psi$ will be satisfied.

The simulation and formal verification are possible in Uppaal. The simulation can be random or user assisted. It is more suitable for verification whether the model corresponds with the real system. Formal verification should confirm that the system has desired properties expressed using the query language. There are many options and settings for verification algorithm in Uppaal. For example we can change representation of reachable states in memory. Some of the options lead to less memory consumption, some of them speed up the verification. But improvement in one of these two characteristic leads to a degradation of the other usually.

For more exact definitions of modeling and query languages and verification possibilities of Uppaal see [2].

## III. Pessimistic protocol two-phase-locking

In this section we suggest one model of pessimistic concurrency control protocol. Of course, it is not the only one possible.

Two-phase-locking protocol is based on data locks. Before access to data the transaction must have a lock. All locks granted to a transaction are released after all operations of this transaction are executed. There are two types of locks—for read and write. The first is used for the operation select and the latter for update, delete and insert. Either one write lock or several read locks can be on a particular record (for simplicity, in our model will be only one read lock allowed for one record). If a transaction can not get a lock for a request it is placed in a queue of this record. After an existing lock is released, a new lock is granted to the first transaction in the queue.

The suggested model consists of several timed automata created using two templates. One type of automata represents data records in a database. The template is shown on the Figure 2. Each record automaton has an integer ID stored in `rec_id`. There are three locations corresponding to two types of locks and to an unlocked state. Channels `rd_ch[x]` and `wrt_ch[x]` are used for requests for read and write locks on record $x$. Channel `rls_ch[x]` is for release (unlock) request.

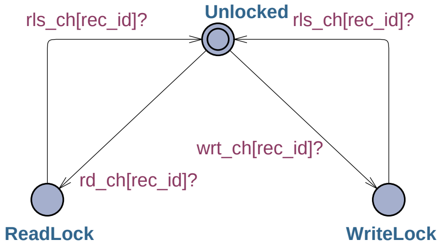


Fig. 2. Automaton representing a record in a database

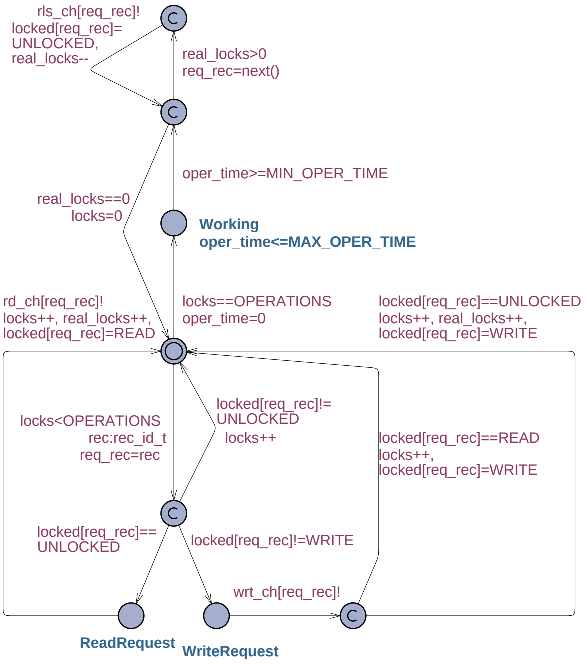


Fig. 3. Transaction automaton for two-phase-locking protocol

The second template shown on the Figure 3 is intended to create automata representing active transactions in the system. In V4DB is a number of active transactions bounded (predispatcher module of RTDBMS holds the queue of incoming transactions and passes them to a dispatcher in such a way that it avoids overloading). So it is possible to represent one active (i.e. currently in execution) transaction as one automaton. After successful end of a transaction the same automaton represents some other transaction.

For simplicity, all transactions are supposed to have the same number of operations (given by a constant `OPERATIONS`). Each operation accesses one record (i.e. needs one lock). A type of operations and an accessed record is for a real RTDBMS in fact random because it is determined outside the RTDBMS. We do not need to model concrete operations, only locks. The record is chosen nondeterministically using select `rec:rec_id_t`. The operation is then immediately (due to a committed location) chosen nondeterministically by using one of three possible edges. If a transaction owns the demanded type of a lock on the accessed record, it does not asks the lock again. If it has only a read lock, it can ask change to a write lock. In the array `locked` is stored the information about owned locks, variable `locks` contains the number of operations for which locks are gained and `real_locks` the number of records locked by this transaction.

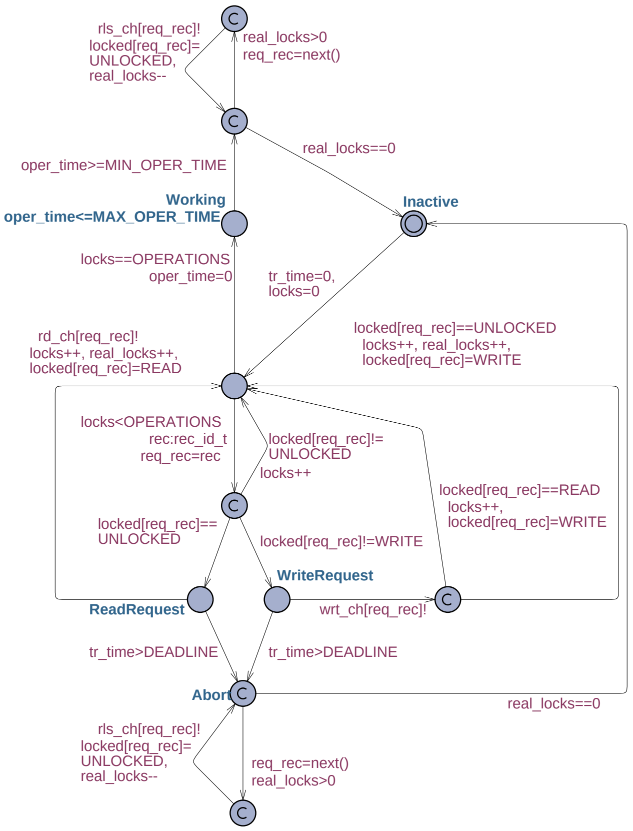


Fig. 4. Transaction automaton for modification of two-phase-locking protocol

If the transaction has all necessary locks, the automaton is in a location `Working`. This represents execution of database operations. The time spent in this location is bounded by constants `MIN_OPER_TIME` and `MAX_OPER_TIME`. After the execution, all locks are released instantly (using committed states and edges between them).

The described model simulates basic variant of 2PL protocol where a deadlock can arise when some transactions wait mutually for locks granted to other transactions. A small modification where transactions exceeding their deadline may be aborted can solve the problem with deadlocks.

## IV. Modification of a model of two-phase-locking protocol

A template for database record automata remains the same as in the previous model (Figure 2). A changed transaction automata template is shown on Figure 4.

There is a clock variable `tr_time` added. It measures time from the beginning of transaction execution. If a transaction is waiting for a lock and it reaches its deadline (for simplicity same for all transactions given by a constant `DEADLINE`), it can be aborted. This means that all locks previously granted to this transaction are released.

We can use Uppaal to verify that this solution is really sufficient to avoid deadlock. For Uppaal, reachability properties are more suitable. So the formula

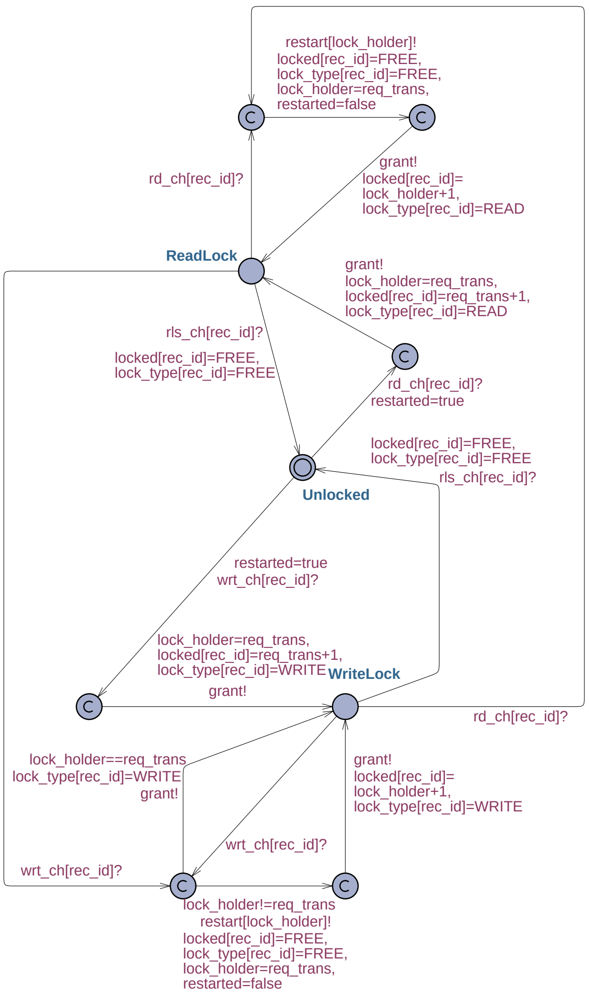


Fig. 5. Automaton representing a record in a database for 2PL-HP protocol

```
E<> deadlock
```

means that deadlock is reachable in the model and this property is not satisfied. Hence it is verified that the system is deadlock-free.

## V. Pessimistic protocol two-phase-locking high-priority

The last modification of our model is for a protocol two-phase-locking high-priority (2PL–HP). If a lock is requested by a transaction with a higher priority the transaction with a lower priority holding this lock may be restarted.

For this model we change both automata templates. A new template for database record automata is depicted on Figure 5.

In the global declarations are defined two arrays – `locked` and `lock_type`. The first one contains information about transactions holding locks for particular records and the latter one contains information about types of particular locks. `lock_holder` is a local variable of one record automaton used for the ID of transaction holding the lock on this record. As almost all is chosen nondeterministically (including the order of activating particular transaction automata), we can model priorities using ID numbers of transaction automata – higher ID means higher priority.

If the automaton is in the location Unlocked, all requests passed through channels rd_ch and wrt_ch are answered immediately through a channel granted and informations about this lock are saved to above mentioned arrays and variable.

If the automaton is in the location WriteLock or ReadLock and a new request arrives, it has to restart a transaction holding the lock (priorities are checked before the request in a transaction automaton). Restarted transaction $x$ is contacted using a channel restart[x]. Then the lock is granted to a requesting transaction using channel grant. If a write lock is requested from the location ReadLock, there is a possibility to grant it without any other activity (except for the change of a type of lock in lock_type array). This is done when requesting transaction (req_trans) is the same as the current holder of the read lock (lock_holder).

The transaction automata template has to be changed as well. The modified version is depicted on the Figure 6.

There are added edges leading to a new location Restart from all locations where an automaton can be during passing of time. All those edges have synchronization label restart[trans_id]. In this way a transaction (with an ID stored in a variable trans_id) can be restarted anytime by a record automaton. In the location Restart all previously gained locks are released and the waiting transaction with higher priority is notified using global boolean variable restarted. A function next() returns the smallest ID number of a record on which is the transaction actually holding a lock.

Requests for locks are guarded. A requested record (specified by the variable req_rec) has to be unlocked or locked by a transaction with smaller priority. It comes handy to use $0$ (constant FREE is defined as $0$) in the array locked[] for unlocked records and ID of transaction automaton (i.e. the priority of transaction) plus one for a lock holder. Than the guard

```
trans_id+1 > locked[req_rec]
```

is true whenever the lock on req_rec is hold by a transaction with a smaller priority or this record is unlocked.d by a transaction with a smaller priority or this record is unlocked.

As in the previous case, although for this model Uppaal can verify that it is deadlock-free. We can use the same formula

```
E<> deadlock
```

and the answer is negative (i.e. no deadlock is reachable).

Furthermore we can check e.g. if the transaction with the highest priority could be possibly restarted. The number of transaction automata is given using a constant TRANSACTIONS. Hence the greatest ID number (this means priority too) is TRANSACTIONS-1. The formula is

```
E<> Transaction(TRANSACTIONS-1).Restart
```

and it is not satisfied, i.e. this transaction could not be restarted. For all other transactions $x$ the formula

```
E<> Transaction(x).Restart
```

is satisfied.

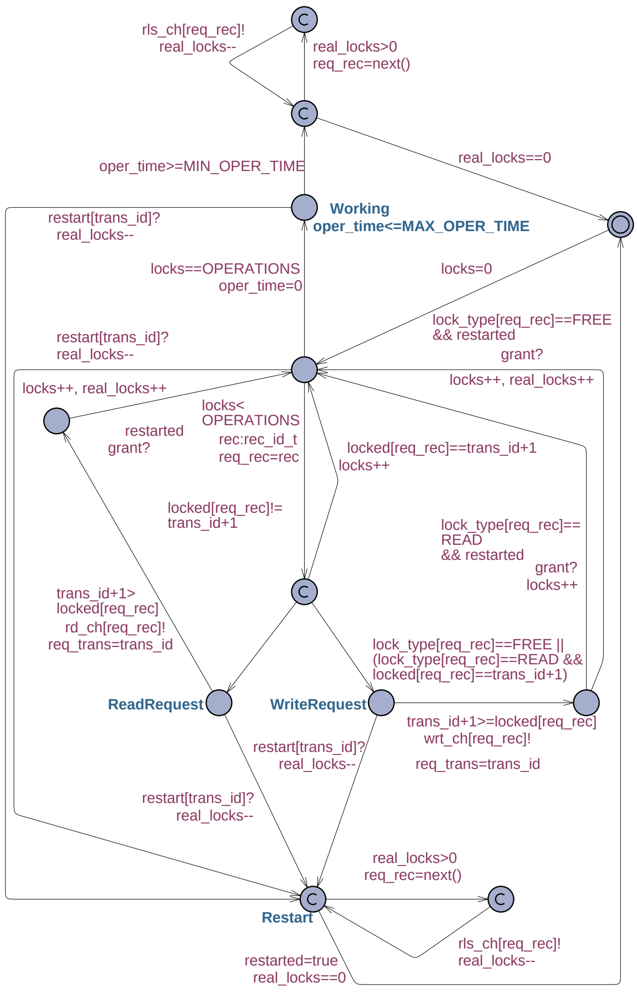


Fig. 6. Transaction automaton for 2PL-HP protocol

## VI. Conclusion

In the previous sections, several timed automata were shown. They form models of three variants of pessimistic concurrency control protocols used in real-time database management systems. Of course, this were not the only possible models. The purpose was to show that some important aspects of the real-time database system such as a concurrency control can be modeled using such a relatively simple model as nets of timed automata are. The models can be extended in many different ways to capture more behavior of those protocols and thus allow many properties to be described as a formula in the logic of Uppaal and then checked using its verification algorithms. Even on presented models (without any extensions or modifications) different properties have been checked and some simple samples of them were presented in this paper.

Some properties can not be expressed using Uppaal's modification of CTL. The possible solution to this problem is to try some other verification tool with other query language.

Other parts of real-time database system or other concurrency control protocols can be modeled too. For example priority assignment algorithms have significant influence on performance database management system. This is our potential future work.

## REFERENCES

[1] Alur, R., Dill, D.L.: Automata for modeling real-time systems. Proc. of Int. Colloquium on Algorithms, Languages, and Programming, volume 443 of LNCS, pages 322-335, 1990.

[2] Behrmann, G., David, A., Larsen, K. G.: A Tutorial on Uppaal. Available on-line at http://www.it.uu.se/research/group/darts/papers/texts/new-tutorial.pdf (September 7, 2007)

[3] Berard, B., Bidoit, M., Petit, A., Laroussinie, F., Petrucci, L., Schnoebelen, P.: Systems and Software Verification, Model-Checking Techniques and Tools. ISBN 978-3540415237, Springer, 2001.

[4] David, A., Amnell, T.: Uppaal2k: Small Tutorial. Available on-line at http://www.it.uu.se/research/group/darts/uppaal/tutorial.ps (September 7, 2007)

[5] Henzinger, T.A.: Symbolic model checking for real-time systems. Information and computation, 111:193-244, 1994.

[6] Król, V.: Metody ověřování vlastností real-time databázového systému s použitím jeho experimentálního modelu. Dissertation thesis. VSB—Technical university of Ostrava, 2006 (in Czech).

[7] Król, V., Pokorný, J., Černohorský, J.: The V4DB project—support platform for testing the algorithms used in real-time databases. WSEAS Transactions on Information Science & Applications, Issue 10, Volume 3, October 2006.

[8] McMillan, K. L.: Symbolic Model Checking. ISBN 978-0792393801, Springer, 1993.

[9] Nyström, D., Nolin, M., Tesanovic, A., Norström, Ch., Hansson, J.: Pessimistic Concurrency-Control and Versioning to Support Database Pointers in Real-Time Databases. Proc. of the $16^{th}$ Euromicro Conference on Real-Time Systems, pages 261-270, IEEE Computer Society, 2004.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 679–685

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Distributed Scheduling for Real-Time Railway Traffic Control

Tiberiu Letia, Mihai Hulea and Radu Miron
Dept. of Automation
Technical University of Cluj-Napoca
C. Daicoviciu St., 15,
40020, Cluj-Napoca, Romania
Email: Tiberiu.Letia, Mihai.Hulea, Radu.Miron@aut.utcluj.ro

***Abstract*—The increase of railway traffic efficiency and flexibility requires new real-time scheduling and control methods. New charter trains have to be added continuously without disturbing the other (periodic) train moves or decreasing the safety conditions. A distributed method to schedule new trains such that their real-time constraints are fulfilled is presented. The trains have timelines to meet and hence the deadlines are extracted taking into account the included laxity. The trains have pre-established routes specifying the stations and required arrival times. The paths containing the block sections from one station to another are dynamically allocated without leading to deadlocks.**

## I. Introduction

### A. Justification of the Problem

RAILWAY networks can solve many of the transportation problems raised by modern society. Railway traffic improvements involve higher flexibility, speed and density. Besides the regular trains (with fixed routes and timetables), new charter trains have to be dynamically accepted without losing the safety and disturbing the other train schedules.

Early conventional railway traffic control has been focused on safety and scheduling train arrival times such that they can be met. Train traffic was of very low density and its efficiency was based on long trains. To avoid train delays, their rates were low and the train speeds were much under their capacities so that the timetable could be fulfilled. Such systems were inflexible and the railway resources were underused.

Railway traffic is described as an emerging network embedded real-time application with long and short reaction time magnitudes. The long durations of event reactions allow the usage of expensive scheduling algorithms that are not accepted in many distributed real-time control applications. The short reaction times involve decisions to be taken under corresponding time constraints.

The railway traffic control system is a dynamic one that operates in an environment with dynamically uncertain properties that include transient and resource overloads, arbitrary arrivals, arbitrary failures and decrease of traffic parameters.

Unlike classical real-time control applications that usually concern only the response times to meet the deadlines, railway traffic involves the reasoning about end-to-end timelines and the reaction to events such that the global traffic system fulfills the time requirements. Despite many uncertainties, the control system is expected to guaranty that all the trains behave according to timelines.

Due to the large dimension of railway networks, the centralized control is not appropriate in the current circumstances because of the need of safety, the communication delays, the complexity of the system and the difficulties to get the right control decisions in time. These are the main reasons for developing autonomous decentralized control systems for railway traffic.

Global train traffic planning is a possible approach of the current problems. A set of trains with their routes and initial departure times is given. The feasible solution provides the train arrival and departure times at the railway stations contained in the given routes. The solution is usually obtained through large system simulation and use of the minimization of different criteria. At this level, train traffic control refers to sending signals such that all the train timetables are fulfilled in all the railway stations.

A train traveling from one point to another involves some dependent real-time activities. The train crossing an interlocking performs an activity directly controlled by the control system. The traveling from one interlocking to another is usually free movement. Some traffic lights can be added to split the long track lines into smaller block sections to increase the track utilization. In this case, a safe policy requires that each block section contains only one train at a time and between each pair of trains a non occupied block section is compulsory. Using Global Positioning System (GPS) and the wireless communication some moving block sections can be implemented. This can lead to higher track utilization, but traffic safety is based on GPS and wireless communication system reliability.

Traffic system goals are:

- to minimize traffic cost;
- to maximize traffic system throughput
- to fulfill train timing requirements;
- to guaranty system safety;
- to minimize fault effects on train schedules and
- to sustain railway maintenance.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

### B. Related Work

The basic principles of railway traffic control are given in [1]. These include the interlocking usage, resource management and dividing the railway network into different parts. The assessment of scheduling is performed by the capability of the schedule to meet the needs of customers and the capabilities of the trains to recover the delays according to their timetables.

The train deviations from the scheduled timetable should be removed during the operation [2].

New trends of train traffic control and management started since 1997 [3]. An autonomous decentralized train control and management system is proposed to attain both the real-time properties for train control such as the real-time traffic and non-real-time properties for train management.

A single delayed train can cause a domino effect of secondary delays over the entire network, which is the main concern of planners and dispatchers [4].

Train scheduling implementations are:

- *off-line scheduling* when all the train arrival times and departure times are calculated before the train starts. The trains behave exactly as they were planned. No unexpected event happens and no new train can appear.
- *on-line scheduling* when the scheduling is performed during the train traffic operation. Some trains have variable delays, unexpected events happen, and new train scheduling requests are required and accepted during the operation.

Some train scheduling approaches are based on:

- distributed artificial intelligence (using trackside intelligent controllers [5]). This kind of allocation of function can optimize the use of resources, reduce complexity and enhance the reliability and availability of the traffic system.
- heuristics methods (as genetic algorithms [6] or ant colony systems [7]). The NP-hard problem complexity with respect to the number of conflicts in the schedule is avoided by generating random solutions and guiding the search.
- auction-based [8]. Each train is represented by an agent that bids for right to travel through a network from its source to destination.
- interactive scheduling [9]. Interactive applications are used to assist planners in adding new trains on a complex railway network. It includes many trains whose timetables cannot be modified because they are already in circulation.

An improvement can be obtained using the GPS and wireless communication between train engine and local control center [10]. Some distributed signal control systems based on the Internet technology are also used [11].

Formal development and verification of a distributed railway control system are performed applying a series of refinement and verification steps [12].

The distributed train scheduling problem has some similarities with distributed software job scheduling [13], [14]. Both have to fulfill real-time constraints relative to finishing time, communication requests and resource management. The concept of collaborative scheduling is also applicable. The problem of the railway interlocking scheduling has some common features with independent scheduling of each node of a distributed software system. Each node constructs its local schedule using only local information. The lack of global information makes it impossible for a node to make a globally optimal decision. Thus it is possible for a node to make a scheduling decision that is locally optimal in terms of the utility that can be accrued to the node, but compromises global optimality. The collaborative scheduling is a paradigm for systems that can withstand its large overhead.

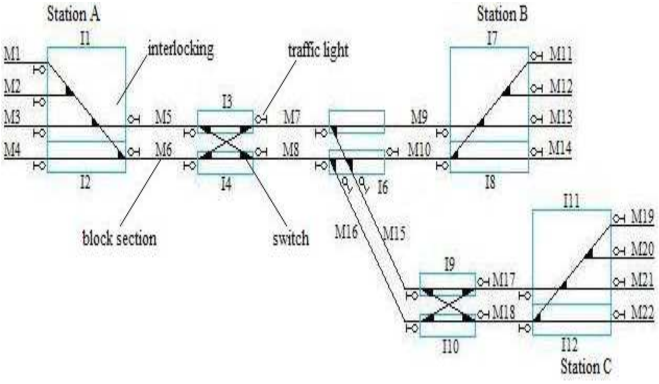


Fig. 1. The railway network structure

## II. Structure and Objectives

### A. Railway Structure

Figure 1 represents a railway network between three stations. On the graph are represented the traffic lights and the switch points. The interlockings are marked by I1, , I12. M1,, M22 denote the block sections controlled by managers. The train movements are controlled by traffic lights and switch points.

An *interlocking* is an arrangement of neighbor interconnected sets of (switch) points and (traffic light) signals such the train movements through them is performed in a proper and safe sequence.

Generally, a *train schedule* is a designation of train description, day, route, speed, arrival and departure times of a train. The train schedule also contains the station dwell times. Some other train stops are required if the necessary track lines are not available when the trains reach the interlocking.

Figure 2 shows a train trajectory with a variable laxity. The notations are:

- *ea* for earliest arrival time;
- *la* for latest arrival time;
- *er* for earliest release (exit) time and
- *lr* for latest release (exit) time.

The objective is to schedule the train move such that it arrives at the next (destination) point between earliest and latest arrival time.

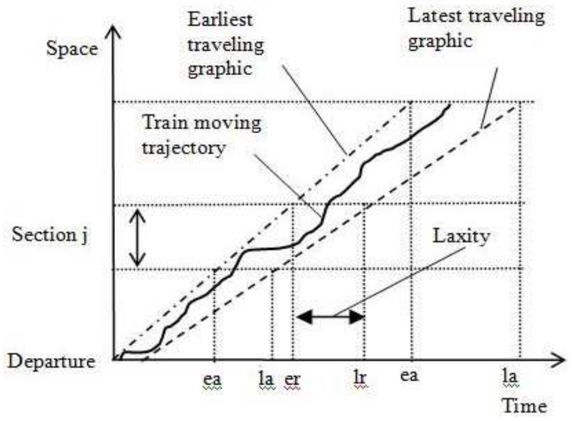


Fig. 2. Train traveling diagram with variable laxity time

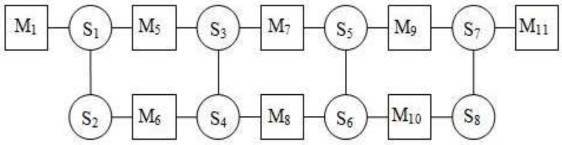


Fig. 3. Distributed scheduling structure

The control decisions take as a target to lead the train such that it reaches the earliest arrival time. The performance evaluation of the system behavior considers that the control fulfills the requirements if all the trains arrive before their specified latest arrival times. If the laxity time is not consumed during a block section or in front of a switch it is added and available to the next sections.

A similar case can be constructed such that the laxity time remains fixed on each section. The train move on a section has to recover the laxity time consumed on the previous one.

In both cases of policy decisions, the laxity time provides the deadline of train behavior together with the earliest arrival time.

### B. *Traffic Control Objectives*

The current study refers to finding a train path starting at a given time, the schedule and the control between two neighbor stations such that the global planned times are fulfilled. This means to find a path from one railway station platform (or block section) to the neighbor railway station platform and the necessary resources (block sections and interlockings). The train traffic control between the neighbor stations means to apply the schedule on-line. The traffic light and switch (point) signals are applied according to the trains current positions and schedules.

## III. Distributed Scheduling

### A. *Architecture of Distributed Scheduler*

Figure 3 represents the software components involved by a train traveling path scheduling. The section managers are denoted with $M_1, \ldots, M_{11}$. $S_1, \ldots, S_8$ represent the interlocking schedulers.

The distributed scheduling is performed by collaboration of train agents, interlocking schedulers and block section managers. The agents have information about the possible paths and expected timetables. For some (charter) trains an acceptable solution is such that the mentioned trains reach the destination as soon as possible.

The interlocking schedulers allocate their resources on-line when the trains approach the interlocking; meanwhile the block section managers allocate the resources off-line (before the trains start their travel from one station to another). An agent asks a scheduler to reserve its controlled interlocking for a specified duration during a given time interval. The scheduler grants it only if the requested task does not delay unacceptably the already scheduled trains, such that the last ones miss their deadlines.

### B. *Agent Behavior*

The train agent's goal is to get a path that fulfills the timing requirements from a railway station platform or block section to another neighbor station.

The agent has to solve a local problem defined by the current train position, departure time, next station block section and arrival time. The planned duration has included, besides the necessary moving time, a laxity time used to compensate the waiting (delay) times involved by crossing of interlockings.

There are two train agent behaviors (approaches):

- In the first one the train agent asks all the schedulers and managers of the possible paths (*listOfPaths*) from the departure station to the neighbor station to accept the train moves. The train agent chooses the best schedule analyzing the schedulers and managers responses.
- In the second one the train agent demands the move specifying the train parameters only to the first scheduler. This is responsible further on to get all the possible paths from departure to neighbor destination. The train agent gets the possible schedules and chooses the best of them. It announces the neighbor scheduler about the chosen path. The neighbor scheduler announces further its neighbor involved components in the path about the firm reservation.

The first approach involves a transaction where the train agent (algorithm) takes a list of possible paths, the start time and the laxity. Using the schedulers and the managers it fills the list of paths with times. Finally, it chooses the best path and announces the scheduling participants about that.

The following notations are added:

- *is* for train input speed at the arrival time at the entrance in the interlocking;
- *os* for train output speed at the exit from the interlocking;
- *st* for train start time;
- *Lx* for laxity time;
- *C* for train worst case crossing time of the interlocking;
- *dd* for absolute deadline.

*Agent algorithm* (first approach):

```
input: trainParameters, trainRoute, Lx;
input: currentStation, listOfPaths;
input: listOfSchedules;
output: trainSchedule;
Initialization: listOfSchedules = extend(listOfPaths);
currentSection=getCurrentSection();
ea=st; la=st+Lx;
is=0;
for all paths from listOfPaths do
    choose an unvisited path;
    while (next) choose the next component as next; do
        (er,lr,os)=next.request(trainParameters,ea,la,Lx, is);
        if (er==0) then break;
        else ea= er; la=lr; is=os; fill in the listOfSchedule;
    end while;
end for;
choose the bestPath from the listOfSchedule;
notify all the participants;
return bestPath as the train schedule;
```

The laxity of traveling from one station to the neighbor station is distributed uniformly to all the paths sections. During traveling from one section to another, if the laxity was not consumed it can be added to the next section.

### C. Scheduler Behavior

Each interlocking is controlled by a scheduler. This can respond to another software component if a new train $T_k$ can be scheduled during a given time interval $[T_k.ea, T_k.la]$ (where ea is the earliest arrival time and la is the latest arrival time of the train $T_k$ at the entrance of the interlocking) without unacceptably delaying the already scheduled trains. If any train is scheduled such that the finishing time of crossing the interlocking is longer than the deadline, then the schedule of the train set is not feasible. A new train can be added to be moved through interlocking only if the schedule of the all train sets with the arrival time intervals overlapping is feasible.

The train $T_k.dd$ deadline of crossing through interlocking is given by:

$$T_k.dd = T_k.ea + T_k.Lx \tag{1}$$

*1) Feasibility analysis:* The worst case for the feasibility analysis is when all the trains of a set arrive simultaneously as close as possible to their deadlines.

Let $t_x$ be the latest time when the trains of a given set can arrive at the same time.

$$t_x = \sup_t \bigcap_k T_k.AI \tag{2}$$

where $T_k.AI = [T_k.ea, T_k.la]$ is the arrival time interval of the train $T_k$ at the entrance of an interlocking. The time $t_x$ is the latest arrival time of any train contained in the intersection of arrival intervals of the considered set.

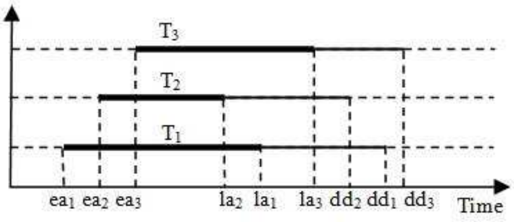


Fig. 4. Train arrival intervals

Figure 4 shows the arrival intervals of three trains $(T_1, T_2, T_3)$ with overlapping arrival time intervals. For this example $t_x = la_2$.

Jackson's rule states: ”Given a set of n independent tasks, any algorithm that executes the tasks in order of nondecreasing deadlines is optimal with respect to minimizing of the maximum lateness [15].”

The feasibility test is:

$$\forall i = 1, ..., n; \sum_{k=1}^{n} T_k.C \leq T_i.dd \tag{3}$$

The previous formula is used off-line to feasibility test analysis. This test has to be applied for all train sets that overlap with the new train added to schedule.

The scheduler uses for on-line traffic control the earliest deadline first (EDF) algorithm. That means if more than one train arrives at the same time, the train with the earliest deadline gets first the right to cross the interlocking. Taking into account that the train deadlines are fixed, the algorithm can be applied on-line using fixed priorities.

The list *scheduledQueue* contains elements with the attributes: *trainIdentifier* $T_i, T_i.ea, T_i.la, T_i.dd, T_i.C.$

For evaluation of the worst loading of an interlocking by a train set with the overlapping arrival intervals, the following formula can be used:

$$load = \frac{\sum\limits_{k=1}^{n} T_k.C}{\sum\limits_{k=1}^{n} (T_k.dd - t_x)} \tag{4}$$

For the reason of robustness a path with smaller load factors of the contained interlockings is preferred. On the other side, if the load is small, there is a greater possibility to obtain a feasible scheduling if a new train agent demands the move through interlocking.

*2) Scheduller Algorithm:* The following notations are added:

- *Cmin* for the minimum crossing time of the interlocking;
- *Cmax* for the maximum crossing time.

*Scheduler grant algorithm of an interlocking:*

```
input: trainParameters;
input: ea, la, Lx, is;
input: scheduledQueue;
input: interlockingParameters;
```

```
output: er, lr, os;
output: feasibility, load;
Initialization: load=0; lr=0;
calculate the best case crossing time Cmin using
  trainParameters and interlockingParameters;
er = ea + Cmin;
calculate the worst case crossing time Cmax and os
  as the maximum speed at the exit of the interlocking
  using trainParameters and interlockingParameters;
find all the train sets with arrival intervals that
  overlap with the arrival interval of the new train;
for all train sets do
    calculate the worst case arrival time $t_x$ of the
      trainSet using formula (2);
    dl =add the worst case crossing time Cmax of
      all trains from trainSet;
    temp = la + Cmax + dl;
    if the formulae (3) are fulfilled with C = Cmax
        then feasibility = true;
        else feasibility = false; break;
    ld = $(\Sigma_i T_i.C)/(\Sigma_i (T_i.dd - t_x)$;//formula (4)
    if (load $\prec$ ld) then load=ld;
    if (lr $\prec$ temp) then lr = temp;
end for;
return er, lr, os, feasibility, load;
```

### D. The Scheduling Improvement

In the presented scheduling algorithm, if a train with lower priority arrives with a very short duration earlier than a train with higher priority, the first one gets the right of crossing. This is inconvenient if the train global priorities express the operator's desires that some trains have to use the interlocking before the others when they arrive almost in the same time.

An algorithm improvement can be: if a lower priority train arrives before a higher priority train, and the first train cannot cross the interlocking before the higher priority train arrival, but the first one can be delayed without missing the timing constraint, the interlocking has to be blocked until the higher priority train arrives and then the EDF algorithm is applied.

An oracle construction can be performed based on GPS or installing detectors on the block sections and estimating the arrival time at the interlocking based on the train current speed. That leads to know in advance the train arrival times during a specified period of time.

Let $T_i.at$ be the arrival time of the train $T_i$ and $B$ the blocking time of the interlocking until the higher priority train arrives. The test of scheduling feasibility is:

$$\forall i = 1, ..., n; T_i.at + B + \sum_{k=1}^{n} T_k.C \leq T_i.dd \qquad (5)$$

The trains can accept different blocking times given by the formula:

$$B_i = T_i.at - T_i.at - \sum_{k=1}^{n} T_k.C \qquad (6)$$

TABLE I
BLOCK SECTION - STATE TABLE

| Time | Solicitor | State |
|---|---|---|
| 0 | $Train_x$ | occupied |
| 1 | $Train_y$ | reserved |
| 2 | $Train_y$ | reserved |
| 3 | - | free |
| 4 | $Train_z$ | requested |
| 5 | $Train_z$ | requested |
| ... | ... | ... |

This acceptable blocking time depends on the train arrival time and its deadline.

Trains with higher priorities usually have higher speed and the proposed improvement involves that a higher priority train can cross the interlocking without waiting. That makes possible that a higher priority train needs shorter laxity time such that the feasibility scheduling test to be fulfilled. This improvement can be used to diminish the unexpected delay of a train due to some faults.

### E. Manager Behavior

The resource manager has the task to reserve on the train agent's request the block section and to maintain the current state of the resource. A block section could have the following state: free, requested, reserved and occupied in every minute. The manager gets information from sensors about occupancy and clearance of the section. The section state is updated at every minute.

The resource manager keeps the Block Section - State Table with reserved periods of the resources for each train.

The train agent asks the reservation calling the method:
*request(trainParameters, ea, la, Lx, is)*

The manager has information about section length and maximum accepted speed. It calculates the necessary time to move from one end to another and reserves an extra *Lx* time. If it is not able to perform the reservation, the manager reserves zero length time intervals.

*Manager request algorithm:*

```
input: trainParameters;
input: ea, la, Lx, is;
input: sectionSpeed, sectionLength;
output: er, lr, os;
determine the train speed sp;
calculate the moving time mt;
er = ea + mt;
lr = mt + Lx; // calculate the later release of the resource
if(the resource is free between ea and lr) then
    mark on the Block Section State Table the attempt of
      reservation for trainID;
    return er, lr, os=sp;// respond with the latest
      // release time and the output speed;
else return er=lr=0, os=is;
```

Agent confirmation call is performed by the method:
*confirm(trainID, ea, lr)*

That reserves firmly the necessary resources and releases the resources attempted to be acquired, but not necessary for the chosen path.

## IV. Implementation and Tests

Two approaches were used to test the scheduling algorithms. One uses the implementation of the proposed algorithms (based on real-time scheduler) and the other uses an implementation based on genetic algorithm. Both approaches use the same railway network model and have the same set of trains already scheduled. A new train schedule is required. Meanwhile the real-time scheduler uses the earliest arrival time as a target and the deadlines only for scheduling feasibility test; the evolutionary system has the goal to obtain the shortest traveling times having the arrival times between earliest arrival times and deadlines.

### *A. Genetic Algorithm for Train Traffic Scheduling*

A *path* from a platform (or a block section) of station A to a platform (or a block section) of station B consists of a sequence of linked elements (interlockings and block sections) used by a train for moving from departure to destination.

The solving of the *scheduling problem* using a *genetic algorithm* has the goal to find the best path and the train speeds on the contained (path) elements. As a consequence, a solution is a pair *(path, set of speeds)*.

A *train schedule* (departure time, path, set of speeds) is *viable* if the train reaches the destination and its trajectory does not overlap any time and any element of the trajectory of any other train from the given train set.

A *train schedule is better* than any other if, starting at departure time and following the solution (path, set of speeds), the train reaches the destination at a time closer to arrival time (if there is given an arrival time), or earlier (if no arrival time is specified) than the time obtained with other solutions.

Between two stations there are a limited number of paths.

*1) Individual coding:* An individual codifies all the paths from departure to destination and the train average speeds on all involved elements. This codification is implemented on a matrix with the number of lines equal with the number of possible paths, and the number of columns equal with the maximum number of elements of any of the paths from departure to destination. As a consequence, each matrix line corresponds to a path. The elements of the line describe the train average speeds on the path elements. Due to the possibility that the path element numbers differ from one path to another, some elements on the right-side of the matrix could not correspond to real train speeds.

*2) Individual evaluation:* Using the train departure time, departure block section and individual codification, the train *traffic simulator* determines for each path of an individual the arrival times. The railway net traffic could contain other trains already scheduled and their schedules are not acceptable to be modified. If the trajectory of an already scheduled train overlaps at the same time the trajectory of the train attempted to be scheduled, the value of the fitness function corresponding to this path is drastically penalized. The evaluation of an individual is given by the *fitness function* that in this case is the weighted sum of the schedules (path, set of speeds) evaluations.

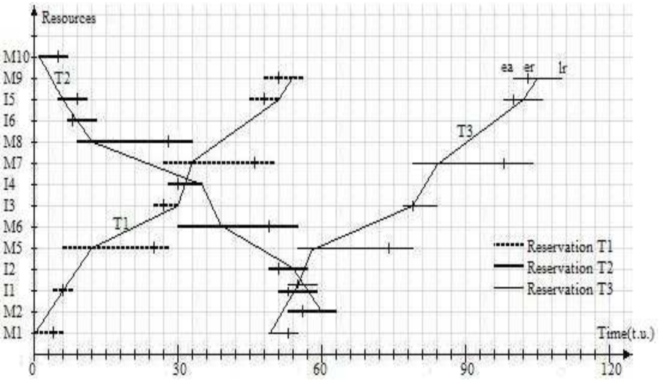


Fig. 5. Example of scheduling

*3) New individual creation:* The solution search using the genetic algorithms is performed by individual creation and evaluation. A new individual creation is obtained by:

- *Mutation.* An individual line (i.e. a path) and an element (i.e. the train speed on an element) of it are randomly chosen. The value of the element is randomly modified taking into account the specified speed limits.
- *Crossover.* Two individuals are chosen. A randomly chosen matrix column is used to cut the individuals' matrices in two parts. Two new individuals are constructed using parts from different matrices.

*4) Individual's selections:* Genetic algorithms work with populations of individuals. The selection of individuals that survive from one generation to another is obtained using the fitness function. The individuals with higher values of fitness functions have higher chances to survive.

The *solution of the scheduling problem* is chosen by taking from all the individuals the best value of the pair (path, set of speeds).

### *B. Solution Comparison*

The solutions obtained using the distributed scheduling algorithms and the genetic algorithm are represented in Figure 5.

The solution given by the genetic algorithm for the traveling of three trains is represented by continuous lines. On the figure are also drawn the block section reservations provided by the real-time scheduler. Each horizontal line describes the mentioned values of *ea*, *er*, and *lr*. The interlocking *I1* is concurrently demanded by two trains (T2 and T3). The genetic algorithm solution avoids the simultaneous use of the interlocking by delaying the T2 train.

## V. Conclusion

The proposed scheduling method does not lead to deadlock due to advance resource reservation. Comparing the performance of the proposed real-time scheduling algorithms with the genetic algorithm performance, the first is lower but needs much smaller computation power (memory and time). The proposed method can provide deterministic time to get the solutions. It also has the advantage to be finally applied on-line and so it is able to diminish the variations of the train arrival times. The proposed method can be used to design the railway networks such that to be capable of providing a specified throughput with real-time features.

## References

[1] J. Pachl, "Railway Operation and Control," VTD Rail Publishing, Mountlake Terrace WA 98043 USA, 2004.

[2] J. Tornquist and J. A. Persson, "Train traffic deviation handling using tabu search and simulated annealing," *Proceeding of the 38th Annual Hawaii International Conference on System Science,* 2005, p. 73a.

[3] S. Shoji, A. Igarashi, "New trends of train control and management systems with real-time and non-real-time properties," *Proceedings of the 3rd International Symposium on Autonomous Decentralized Systems (ISADS'97),* 1997, pp. 319–326.

[4] R. M. P. Goverde, "A delay propagation algorithm for large-scale scheduled rail traffic," *Preprints of 11th IFAC Symposium on Control in Transportation Systems,* Delft, Netherlands, 2006, pp. 169–175.

[5] T. Tao, "A train control system for low-density lines based on intelligent autonomous decentralized system (IADS)," *Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03),* 2003.

[6] A. Higgins and E. Kozan, "Heuristics techniques for single line train scheduling," Journal of Heuristics, 3, Kluwer Academic Publishers, 1997, pp. 43–62.

[7] K. Ghoseri and F. Merscedsolouk, "ACT-Ts: Train scheduling using ant colony system," Journal of Applied Mathematics and Decision Science, 2006, pp. 1–28.

[8] D. C. Parkes and L. H. Ungar, "An auction-based method for decentralized train scheduling," Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, Canada, 2000, pp. 43–50.

[9] A. Lova, P. Tormas, F. Barber, L. Ingolotti, M. A. Salido and M. Abril, "Intelligent train scheduling on high-loaded railway network," Algorithmic Methods for Railway Optimization, Springer, Berlin, 2007.

[10] A. Zimmermann and G. Hommel, "A train control system case study in model-based real-time system design," *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03),* Nice, France, 2003, p118b.

[11] Y. Fukuta, G. Kogure, T. Kunifuji, H. Sugahara, R. Ishima and M. Matsumoto, "Novel railway signal control system based on the internet technology and its distributed control architecture," *Proceedings of the Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07),* 2007.

[12] A. Hauxthausen and J. Peleska, "Formal development and verification of a distributed railway control system," *IEEE Trans. on Software Engineering,* Vol. 26, No. 8, 2000, pp. 687–701.

[13] S. F. Fahmy, B. Ravindran and E. D. Jensen, "On collaborative scheduling of distributable real-time threads in dynamic, Networked Embedded Systems," *Proceedings of the 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC),* 2008, pp. 485–491.

[14] S. F. Fahmy, B. Ravindran, and E. D. Jensen, "Scheduling distributable real-time threads in the presence of crash failures and message losses," ACM SAC, Track on Real-Time Systems, 2008.

[15] G. C. Buttazzo, "Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications," Second edition, Berlin: Springer-Verlag, 2005.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 687 – 693

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Wireless Sensor and Actuator Networks: Characterization and Case Study for Confined Spaces Healthcare Applications

Diego Martínez
Universidad Autónoma de Occidente
Km. 2, Cali - Jamundí, Colombia
dmartinez@uao.edu.co

Francisco Blanes, José Simo, Alfons Crespo
Polytechnic University of Valence
Camino de Vera S/N Valence, Spain
{ pblanes, jsimo, acrespo }@ disca.upv.es

***Abstract*—Nowadays developments in Wireless Sensor and Actuators Networks (WSAN) applications are determined by the fulfillment of constraints imposed by the application. For this reason, in this work a characterization of WSAN applications in health, environmental, agricultural and industrial sectors are presented. A case study for detecting heart arrhythmias in non-critical patients during rehabilitation sessions in confined spaces is presented, and finally an architecture for the network and nodes in these applications is proposed.**

## I. Introduction

Currently, there is a great interest in developing applications for monitoring, diagnosis and control in the medical, environmental, agricultural and industrial sectors, to improve social and environmental conditions of society, and increasing quality and productivity in industrial processes. The development of Wireless Sensor and Actuators Networks (WSAN) applications will contribute significantly to solve these problems, and facilitate the creation of new applications.

Some applications which can be developed using WSAN are:

- Medical Sector: economic and portable systems, to monitoring, recording and analyzing physiological variables, from which it is possible to indicate the status of a patient and detect the presence or risk of developing a disease. As well, developing systems for the detection and analysis of trends in the daily behavior of patients, contributing to timely detect the presence of a health problem, and provid ing an economically viable solution to patient care in societies where the old population is great.
- Environmental Sector: continuous systems monitoring of species in dangerous extinction, monitoring and detection of forest fire systems, etc.
- Agricultural sector: detection systems, microclimates monitoring and pest control, to reduce the use of agrochemicals and make an optimal control of pests; optimal use of water in irrigation systems, etc.
- Industry: economic systems and easy installation for monitoring, diagnosis and control of plants and industrial processes.

Some of the currently technological challenges in WSAN development are [1], [2], [3], [4], [5]:

- It is necessary to develop detailed models of the system components (hardware and software tasks, task scheduler, medium access control and routing protocols), in languages that allow correct specifications and the subsequent analysis of information processing, reachability, security, and minimum response time application, enabling analysis of end to end deadline in real time applications.
- Task scheduler and medium access control and routing protocols proposed in this area are mostly focused on the optimization of a single critical parameter of the application, which often affects considerably the performance of the others. Therefore it is necessary to create new cooperation forms between these levels of the application architecture, in order to take the most appropriate decisions for the system reconfiguration in relation to the application's quality of service (QoS). Additionally, current proposals consider QoS parameters directly linked to conventional parameters of the operation and communication between computers but not to particular application requirements, so do not allow achieving optimal performance in applications.
- It is necessary to develop analysis strategies for performance and stability of signal processing and control algorithms in this area, in order to guide the design towards a co-design methodology to develop the processing algorithm and the implementation of computer architecture, allowing to compensate the sampling period changes and jitter effects, and optimize other parameters such as power consumption.

The previous paragraphs show how these developments are determined by the fulfillment of constraints imposed by the application, such as energy consumption, limited computing power, coverage of large areas and real time deadlines, etc. For these reasons, in this work a characterization of WSAN applications for health, environmental, agricultural and industrial sectors is presented; then a case for detecting heart arrhythmias in non-critical patients during rehabilitation sessions in confined spaces is presented, finally an architecture for the network and nodes in these applications is proposed.

The article is organized as follows, section 2 shows a classification and characterization of applications in health, environmental, agricultural and industrial sectors, section 3 presents the case analyzed, in section 4 a proposal for the nodes architecture is presented, the network architecture and its simulation results are presented in section 5, finally in section 6 the conclusions and future work are presented.

This work is partially supported by the CYTED D2ARS Project

## II. Classification of Applications

During the classification was detected that different application sectors share similar characteristics from a technological point of view, for this reason the classification and characterization was developed in five types of application rather by sectors [6], [7], [8], [9], [10], [11], [12], [13], [14].

- Type 1 applications are characterized by measuring sampling periods from one second to few hours, and no strict deadlines for the generation of the algorithms results. Additionally, these applications, developed in open spaces, must cover large areas and it is necessary to synchronize measurements in different nodes. Agricultural and environmental applications, designated to measure, record and to analyze environmental variables, primarily belong to this category.

The energy sensors autonomy expects for each node varies from days to months; in some applications, in places without access to conventional energy sources, nodes are equipped with energy transducers like solar cells, which supply energy to the nodes batteries.

- Type 2 applications are developed in confined spaces and have greater computing capacity demanding than applications type 1, although there is not strict response time, either. Because they are in confined spaces and nodes are fixed, there are no restrictions on energy consumption since they can use conventional energy sources; however, the use of wireless networks is justified since it facilitates the installation, adaptability and portability of implementation, in addition to the lower costs of implementation.

Such applications are fined mainly in industrial and agricultural sectors. The sampling periods range from one millisecond, for implementation of diagnostic algorithms, until a second for monitoring and supervision tasks. The diagnostic algorithms do not require continuous operation; therefore the samples can be stored before being processed.

- Type 3 applications. In this category, in addition to measuring and processing data requirements similar to those in applications type 2, grouped applications are required to process images and they are developed in open spaces, some of which are in the agricultural sector for the detection of pests, and environmental sector aimed at detecting fires.

Some of the nodes are mobile and require few hours' energy autonomy, and then restrictions in terms of power consumption are large. At the same time it is also necessary for synchronization of the nodes. While, because of the algorithms used for image processing, the computing capacity, memory size and communication bandwidth requirements are greater than applications type 2.

- Type 4 applications. These applications differ from applications type 3 because they are developed in confined spaces, then the network's coverage is not demanding; energy sensors autonomy expected are also higher, becoming close to one week. Grouped healthcare applications to the detection of diseases are in this category with body area networks (BAN).
- Type 5 applications. In these applications a sample data should be sent every sampling period, and then sampling periods are limited by the minimum interframes time space of data communication protocols. For this category, a range of sampling periods between 50 milliseconds and a few seconds has been selected.

In this category are the applications of industrial control process, which are developed in confined spaces, so the distance between nodes is not big, and there are not restrictions on energy consumption. The deadlines for generating actions are less than or equal to a sampling period, and it is necessary to guarantee end to end deadline. If these constraints are not fulfilled, the control system performance can be degraded significantly, even generating instability in the system, therefore, it is important to synchronize the activities of the nodes that are integrated in the control loop. In addition, to improve the control system performance, it is important to limit the variability in the task jitter.

Table 1 summarizes the characteristics of the applications described. As a strategy to increase reliability in the presence of faults, and optimizing the applications QoS, this proposal also has considered the migration of components between the nodes, which will be reflected in the architecture of the network and nodes .

## III. Arrhythmia Detection Algorithm

Actually cardiovascular problems have the highest mortality rate from natural causes in the world. The great interest in developing devices for clinical detection and continuous monitoring of such diseases, is based on these activities are limited by the information type and the moment that it is caught, so transitional abnormalities can not be always monitored. However, many of the symptoms associated with cardiovascular diseases are related to transient episodes rather

Table 1
Analysis of requirements for each application type

| **Application** | **Computing capacity** | **Memory size** | **Communication bandwidth** | **Location** | **Node Mobility** | **Real-Time** | **Network's coverage** | **Energy autonomy** | **Synchronization** |
|---|---|---|---|---|---|---|---|---|---|
| **Type 1** | Low performance | Low | < 256 kbps | Yes | No | Only measurement | Open space 10 km | Months | Yes |
| **Type 2** | Low performance | Medium | < 256 kbps | Yes | No | Only measurement | Confine space 100 m | There isn't restriction | Yes |
| **Type 3** | High performance | High | 1 Mbps | Yes | Yes | Only measurement | Open space 10 km | Hours | Yes |
| **Type 4** | High performance | Medium | < 256 kbps | Yes | Yes | Only measurement | Confine space 1 km | Days | Yes |
| **Type 5** | Low performance | Low | < 256 kbps | No | No | End to end and Minimum jitter variability | Confine space 100m | There isn't restriction | Yes |

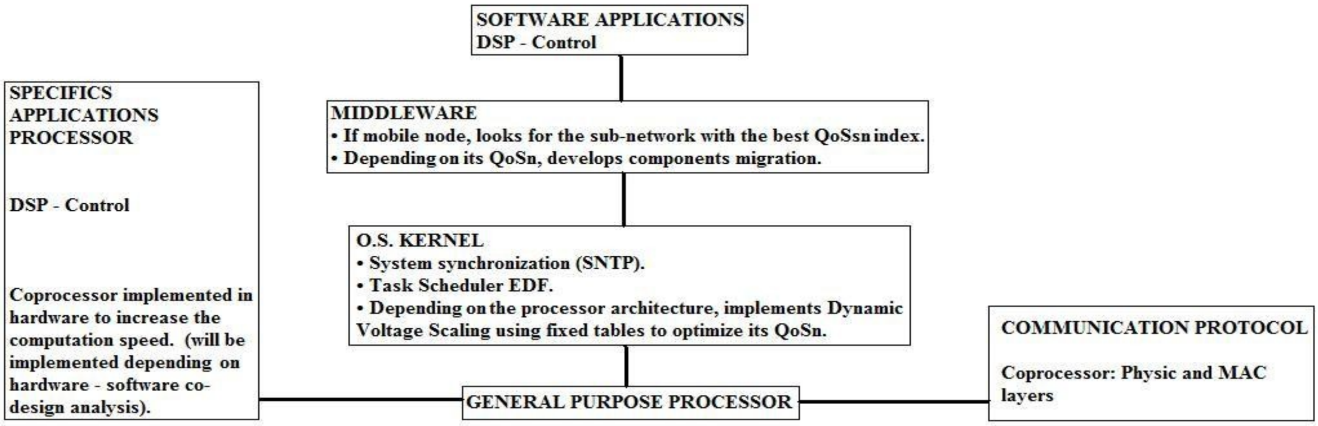

SOFTWARE APPLICATIONS
DSP - Control
SPECIFICS APPLICATIONS PROCESSOR
DSP - Control
Coprocessor implemented in hardware to increase the computation speed. (will be implemented depending on hardware - software co-design analysis).
MIDDLEWARE
• If mobile node, looks for the sub-network with the best QoSsn index.
• Depending on its QoSn, develops components migration.
O.S. KERNEL
• System synchronization (SNTP).
• Task Scheduler EDF.
• Depending on the processor architecture, implements Dynamic Voltage Scaling using fixed tables to optimize its QoSn.
GENERAL PURPOSE PROCESSOR
COMMUNICATION PROTOCOL
Coprocessor: Physic and MAC layers

Figure 2 Nodes architecture

than continuing abnormalities, such as transient surges in blood pressure, arrhythmias, and so on. These abnormalities can not be predicted therefore a controlled supervision analysis is discarded. The reliable and timely detection of these episodes can improve the quality of life of patients and reduce the therapies cost. For this reason in this work a WSAN architecture to address such problems is proposed, this application belong s to type 4 described in paragraph 2.

As an example, the detection of arrhythmias using data from electrocardiogram (ECG) measure, in patients who are moving during a rehabilitation activity in a confined space of 100m x 100m, as a rehabilitation center, was analyzed. The sampling period was selected from the ECG frequency spectrum, which, according to the American Heart Association, has 100Hz harmonics. The greatest amount of relevant information for monitoring and detection of arrhythmias is between 0.5Hz and 50Hz.

When analyzing the ECG frequency spectrum can be established that the relevant components of the signal (QRS complex and waves P and T) are up to 35Hz. Applying the sampling theorem a minimum sampling period of 14ms approximately is necessary, but for practical purposes a period of 3ms was selected.

For the detection and analysis of ECG the Pan and Tompkins algorithm was selected, [15]. The results of this algorithm are used by a maximums detection algorithm, which identifies the time when segments of the ECG wave were presented, figure 1. Subsequently the analysis of the separation time between two R waves, the duration of the QRS segment and the energy of the wave R is developed, which allows detecting the presence of arrhythmias [16].

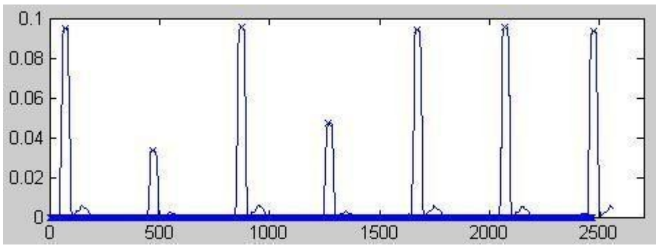


Figure 1 Results of maximums detection algorithm

## IV. Architecture Node

The proposed generic architecture for the network nodes in applications type 4 is presented in figure 2. Its characteristics are:

- The architecture enables the hardware and software components co-design. This feature will allow optimizing the development of distributed application components required for its implementation in hardware and software, getting a balance between cost, power consumption and processing time
- There are fixed and mobile nodes. The latter are linked to the sub-network that guarantees them the best QoS ( $QoS_{sn}$ ) during its movement (less saturated sub-networks).
- Communication between the nodes and local coordinators is done through wireless networks.
- Use an EDF scheduler and dynamic scaling voltage and frequency techniques of the processor to optimize the power consumption [17]. This allows fulfilling the application deadlines, which is supported on statics utilization rate tables for each operating frequency, and periods of execution for each task.
- Update its QoS indexes ( $QoS_{n}$ ); using these indexes and its neighboring ones it is possible to request to another node in its sub-network the migration, creation or destruction of components (some of which are clones of others).

To select a set of architectures for an adequate performance of these applications, the performance of the arrhythmia detection algorithm, presented in section 3, was analyzed on four types of processors currently used to implement nodes in sensor networks: ARM7TDMI, MSP430, PIC18 and MC9S08GB60.

For the analysis , the same operation velocity for each processor was used, 8MIPS. The time necessary to develop the Pan and Tompkins algorithm is presented in Table 2, which was estimated considering the sum of the values of individual functions (derivative, quadratic function and integrator window) in each architecture.

The results show that the ARM architecture requires a lower percentage of utilization, while the PIC architecture needs the highest utilization percentage.

It also was related consumed power by each architecture in active mode ($P_A$) with the respective percentage of utilization

TABLE 2
COMPUTING TIME TO DEVELOP THE PAN AND TOMPKINS ALGORITHM

| Processor | Derivative | Quadratic function | Integrator window | Total computing time | Period [µs] | Percentage of utilization (U) |
|---|---|---|---|---|---|---|
| **LPC2124 – ARM** | 70.2 µs | 142 µs | 280.5 µs | 492.7 µs | 3000 | 16.4% |
| **MSP430F1611** | 191.9 µs | 162.5 µs | 697.8 µs | 1052.2 µs | 3000 | 35% |
| **PIC18F458** | 406.2 µs | 209 µs | 1083.7 µs | 1698.9 µs | 3000 | 56.6% |
| **MC9S08GB60** | 497.2 µs | 332 µs | 707.35 µs | 1536.55 µs | 3000 | 51.3% |

during the implementation of the algorithm, table 3. It can be seen as the ARM7 architecture has a closer performance to the architecture MC9S08GB60; then these two architectures are appropriate for the implementation of the case proposed. The MSP430 architecture presented the best indicator.

TABLE 3
INDICATOR $P_A$*U

| | Utilization percentage (U) | Active Power ($P_A$) [mW] | $P_A$ *U |
|---|---|---|---|
| **LPC2124 – ARM** | 16.4% | 180 | **29.52** |
| **MSP430F1611** | 35% | 19.2 | **6.72** |
| **PIC18F458** | 56.6% | 220 | **124.52** |
| **MC9S08GB60** | 51.3% | 51.6 | **26.47** |

## V. NETWORK ARCHITECTURE FOR CONFINED SPACES HEALTHCARE APPLICATIONS

In figure 3, a generic architecture for network applications type 4, which integrates different types of nodes, is proposed. The approach of a cooperation plan between architecture levels of the application, in order to take the most appropriate decisions for the reconfiguration of the system in relation to the application QoS, can be appreciated. General goals of the architecture are:

- Minimize latencies.
- Optimize power consumption.

The Main Coordinator is responsible for coordinating the complete application. It will have a fixed location, and communication with local coordinators will be supported through wireless or wired links. It develops the following functions:

- Send sync hronization signals to the local coordinators of the sub-networks.

Local Coordinator control s the activity inside the sub-network and develop s some information processing activities, whose architecture is presented in figure 4 and its features are:

- It has a fixed location.
- Sends synchronization signals to nodes in its sub-network.

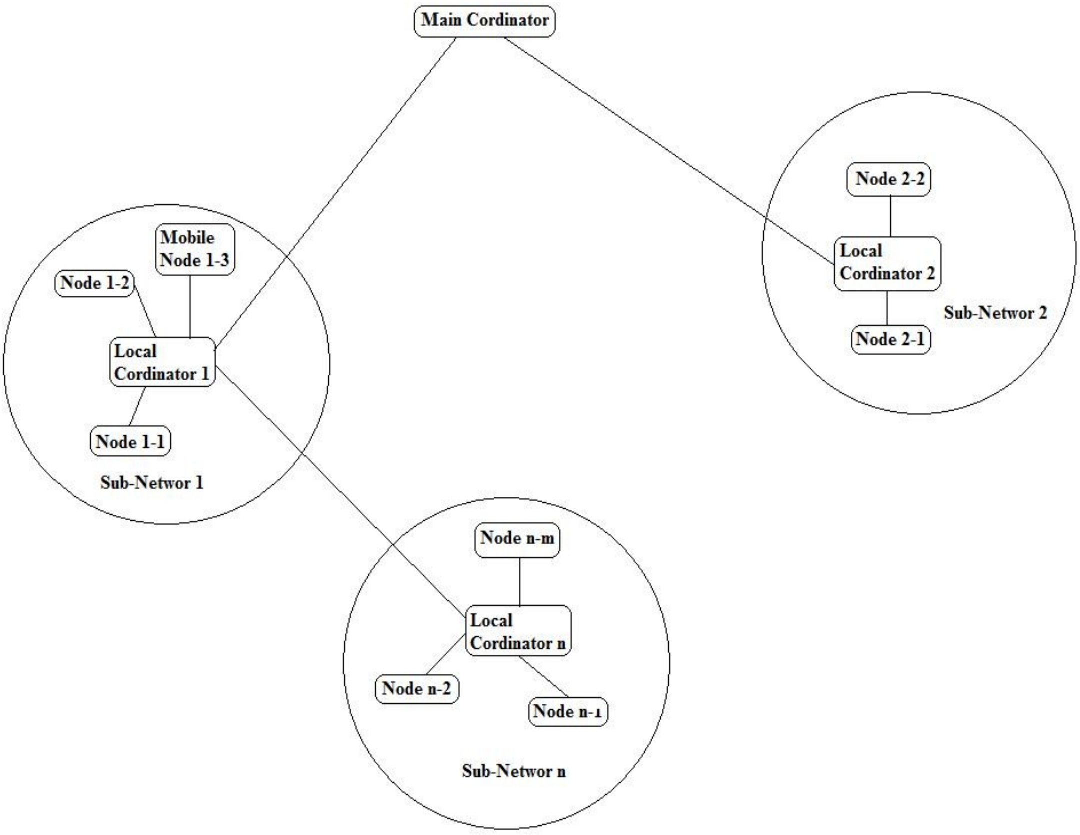


Figure 3 Network architecture

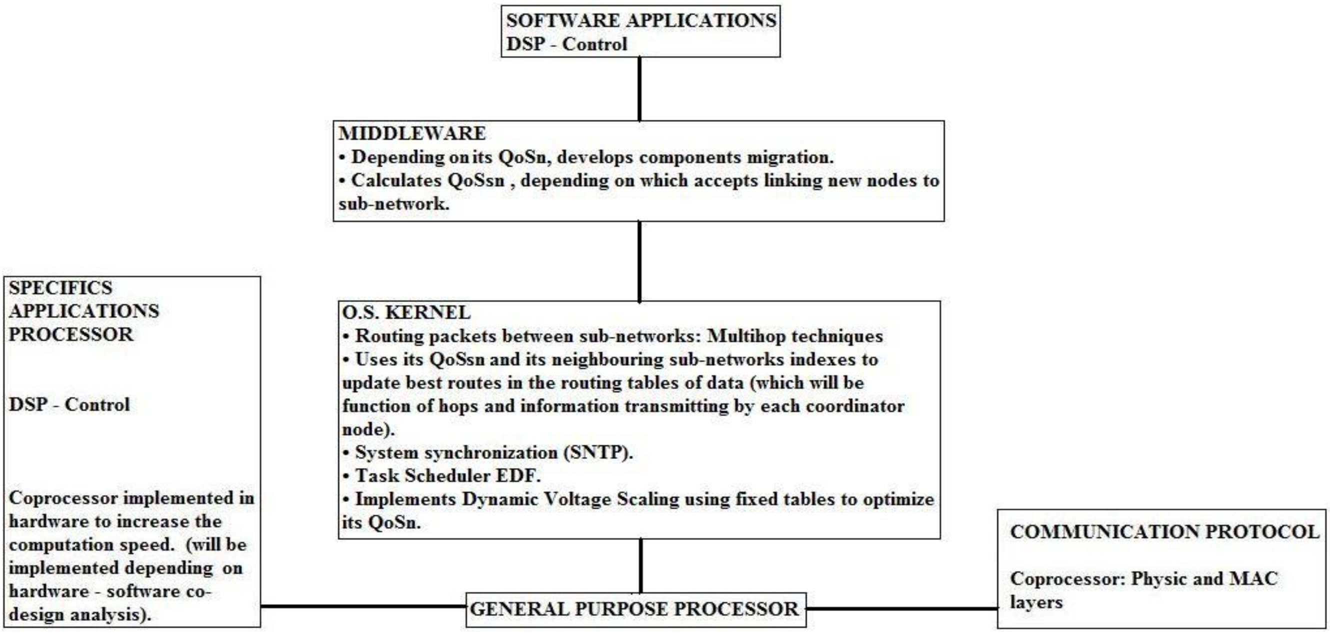

SOFTWARE APPLICATIONS
DSP - Control
MIDDLEWARE
• Depending on its QoSn, develops components migration.
• Calculates QoSsn , depending on which accepts linking new nodes to sub-network.
SPECIFICS APPLICATIONS PROCESSOR
DSP - Control
Coprocessor implemented in hardware to increase the computation speed. (will be implemented depending on hardware - software co-design analysis).
O.S. KERNEL
• Routing packets between sub-networks: Multihop techniques
• Uses its QoSsn and its neighbouring sub-networks indexes to update best routes in the routing tables of data (which will be function of hops and information transmitting by each coordinator node).
• System synchronization (SNTP).
• Task Scheduler EDF.
• Implements Dynamic Voltage Scaling using fixed tables to optimize its QoSn.
COMMUNICATION PROTOCOL
Coprocessor: Physic and MAC layers
GENERAL PURPOSE PROCESSOR

Figure 4 Sub-network coordinator architecture

- Develops routing packets between sub-networks using multihop techniques.
- Distributes QoS indexes of nodes which belong to its sub-network ( $QoS_n$ ) .
- Calculates its sub-network QoS index ( $QoS_{sn}$ = f(quantity of information to be transmitted)), and distributes this value and its neighboring sub-networks indexes (those reached in a single communication hop) between nodes in its sub-network. Depending on which:
  - Accepts linking new nodes to sub-network.
  - Updates best routes in the routing tables of data (which will be function of hops and the utilization percentage -information transmitting- of each router node).

As a first approximation to the proposed architecture, we examined the performance of the case analyzed on the IEEE 802.15.4 protocol. Considerations for the proposed solution to the case are:

- Transmission of the analysis results, from nodes located on each patient to a main node, every 3 s. The data frame consists of 2 Bytes, which contain patient codes and the type of arrhythmia detected.
- After each sending the sender node waits for an acknowledgement (ACK) from next node in the routing path. If there isn't an answer before 100ms the node sends again the information. If after 25 attempts there i s no answer this node change s to a mistake state.
- The communication protocol selected is IEEE 802.15.4. The node distribution is shown in figure 5, which allows cover ing all possible locations of patients considering the specifications of the devices selected to implement the physical layer, CC2420, whose characteristics are:
  - Coverage radio of 30m, and 100m without obstacles.
  - Frequency range of 2.4 - 2.4835 GHz.
  - Supports data transfer rates of 250 kbps.

In the case a network as presented in figure 6 was proposed. It consists of 3 fixed nodes which have no restrictions on power consumption, will receive reports from five patients and rout e the messages to the main node. The fixed devices have fixed identifiers 0, 1 and 2; the main node has the 0 identifier, and devices on every patient have identifiers from 3 to 7.

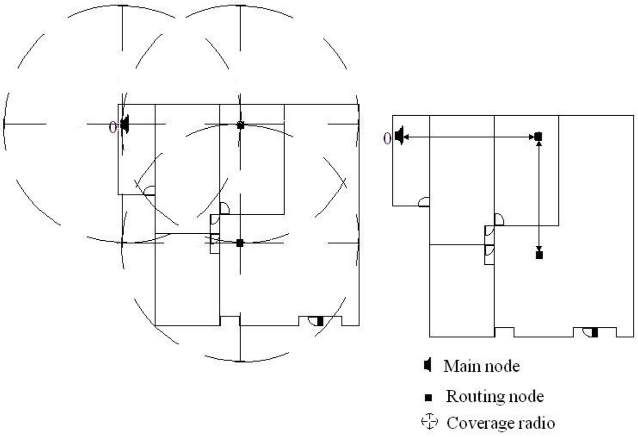


Figure 5 Distribution nodes for case and their coverage

The routing is developed through 1, 2 and 0 nodes, 0 is the network coordinator, each of these nodes form ing a sub-network together with patients, figure 6. The mobile nodes leave and enter the sub-networks continuously changing the configuration and network structures.

The simulation was developed in the TOSSIM tool, and the TelosB platform was selected, including the CC2420 transceiver. Because the characteristics given of the case, with fixed nodes to implement the routing protocols, a routing fixed table algorithm was implemented, it is presented in table 4.

Table 5 Time in sending 2 Bytes from all patients to the main node

| Sender node | Receiver node | Time (s) | Source node | Receiver node | Time (s) |
|---|---|---|---|---|---|
| 6 | 2 | 78.309 | **0** | **1(ack)** $_1$ - 5 ends | .613 |
| 4 | 2 | .320 | 7 | 2 Rtx $_2$ | .684 |
| 7 | 2 | .333 | 3 | 2 Rtx $_2$ | .684 |
| **2** | **6(ack)** | .344 | **2** | **1 Rtx moving 4** $_2$ | .684 |
| 5 | 2 Rtx $_2$ | .355 | **1** | **2 (ack)** | .701 |
| **2** | **1 (moving frame from 6)** | .380 | 7 | 2 Rtx $_2$ | .714 |
| 3 | 2 Rtx $_2$ | .380 | **1** | **0 (moving frame from 4)** | .740 |
| **1** | **2(ack)** | .397 | **2** | **7 (ack)** | .746 |
| **1** | **0 (moving frame from 6)** | .421 | **0** | **1 (ack)** $_1$ - 4 ends | .764 |
| 5 | 2 Rtx $_2$ | .421 | **2** | **1 (moving frame from 7)** | .780 |
| 3 | 2 Rtx $_2$ | .463 | 3 | 2 Rtx $_2$ | .780 |
| **2** | **1 (moving frame from 5)** | .486 | **1** | **2 (ack)** | .792 |
| 7 | 2 Rtx $_2$ | .486 | 3 | 2 Rtx $_2$ | .807 |
| **2** | **5(ack)** | .488 | **2** | **3 (ack)** $_2$ | .825 |
| **0** | **1(ack)** $_1$ - 6 ends | .488 | **2** | **1 (moving frame from 3)** | .860 |
| **1** | **2(ack)** | .500 | **2** | **1 Rtx moving 3** $_2$ | .877 |
| 4 | 2 Rtx $_2$ | .513 | **1** | **0 (moving frame from 7)** | .877 |
| 3 | 2 Rtx $_2$ | .524 | **0** | **1 (ack)** $_1$ - 7 ends | .913 |
| 7 | 2 Rtx $_2$ | .535 | **2** | **1 Rtx moving 3** $_2$ | .929 |
| **2** | **4(ack)** | .547 | **1** | **2 (ack)** | .949 |
| **2** | **1 (moving frame from 4)** | .581 | **1** | **0 (moving frame from 3)** | .979 |
| **1** | **0 (moving frame from 5)** | .589 | **0** | **1 (ack)** $_1$ - 3 ends | 79.013 |

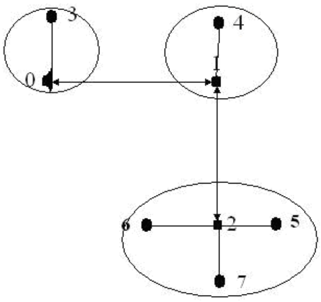


Figure 6 Network structure for case

Table 4 Routing Table

| Source node | Destination node |
|---|---|
| 2 | 1 |
| 1 | 0 |
| 0 | |

In the simulation was considered the most critical case, wh ere all mobiles nodes are connected all time to the farthest sub-network from the main node.

Times obtained in sending 2 Bytes from all patients to the main node (node 0), are presented in table 5. The data$_1$ indicate that transmitting the message from a mobile node to the main node was over; the data$_2$ indicate that the corresponding node has not received the ACK. Times obtained demonstrated that it is possible to fulfill the constraint of 3s, for the transmission of patient status from a mobile node to the main node, which was imposed by the case analyzed.

## VI. Conclusions and Future Work

From the study it can be concluded that developments on specific technologies and applications in this area are still emerging, and developing them will enable the growing of great social impact new applications.

The proposed architecture considers the constraints of application field, allowing to find optimal solutions to the challenges in network and nodes designing, and will facilitate the development and validation of applications. It makes possible too the cooperation between levels of the network architecture to choose between different operations modes depending on QoS indexes.

It is noted as the routing algorithm based on fixed tables supported by IEEE 802.15.4, fulfils the time requirements of these applications. Also as MSP430 architecture presents a good performance for the implementation of the case considered.

The future work proposed is the performance analysis of different task schedulers and routing protocols on the presented architecture, and a cooperation strategy between them to minimize power consumption.

## R eferences

[1] Shivakumar SatrY, S. S. Iyengar. Real-Time Sensor-Actuator Networks. International Journal of Distributed Sensor Networks. January 2005, Pages: 17-34.

[2] Baronti P., Pillai P., Chook V., Chessa S., Gotta A., Y. Fun Hu, “Wireless Sensor Networks: a survey on the State of the Art and the 802.15.4 and ZigBee Standards”, *Computer Communications,* Vol. 30, No. 7, May 2007, Pages: 1655-1695.

[3] Ananthram Swami, Qing Zhao, Yao-Win Hong, Lang Tong. Wireless Sensor Networks. WILEY 2007. Pages: 251-343.

[4] Marrón Pedro José, Minder Daniel, and the Embedded WiSeNts Consortium. Embedded WiSeNts Research Roadmap. November 2006.

[5] Hill Jason Lester. System architecture for wireless sensor networks. *Doctoral thesis.* University of California, Berkeley. 2003.

[6] David Culler, Deborah Estrind y Mani Srivastava. Overview of Sensor Networks. *IEEE Computer,* August, 2004. Pages: 41-49.

[7] B. Son, Y. Her y J. Kim. A Design and Implementation of Forest-Fires Surveillance System based on Wireless Sensor Networks for South

Korea Mountains. *International Journal of Computer Science and Network Security.* Vol. 6 No. 9B, September, 2006.

[8] L. Yu, N. Wang, X. Meng. Real-time Forest Fire Detection with Wireless Sensor Networks. *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing,* 23-26 September. 2005.

[9] P. Klein Haneveld. "Evading Murphy: A sensor network deployment in precision agriculture". *Technical Report.* June 28, 2007.

[10] N. Wang, N. Zhang and M. Wang. "Wireless sensors in agriculture and food industry – Recent development and future perspective". *Computers and Electronics in Agriculture.* Vol. 50, Nr. 1, January 2006, Pages: 1–14.

[11] A. Baggio. "Wireless sensor networks in precision agriculture". *Workshop on Real-World Wireless Sensor Networks.* REALWSN'05. Sweden , June 20-21, 2005.

[12] Benny P L Lo and Guang-Zhong Yang. Key technical challenges and current implementations of body sensor networks. Department of Computing, Imperial College London, UK. http://www.doc.ic.ac.uk/~benlo/ubimon/BSN.pdf.

[13] J. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru and A. Wood. Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges. High Confidence Medical Device Software and Systems *(HCMDSS) Workshop,* Philadelphia, PA, June 2005.

[14] Thomas Norgall, Fraunhofer IIS. "Body Area Network BAN - a Key Infrastructure Element for Patient-Centric Health Services". *ISO TC215/WG7/IEEE 1073 Meeting,* Berlin. May 2005

[15] Pan Jiapu, Tompkins Willis J., A Real-Time QRS Detection Algorithm, *IEEE Trans. Biomed. Eng.,* vol. BME-32, Pages: 230-236, 1985.

[16] Tompkins W J; Webster J G. Desing of microcomputer-based medical instrumentation.New Jersey: Prentice-Hall, 1981. Pages: 396-397

[17] Padmanabhan Pillai, Kang G. Shin. "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems". *Proceedings of the eighteenth ACM symposium on Operating systems principles.* Banff, Alberta, Canada, 2001. Pages: 89 – 102.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Real-Time Service-oriented Communication Protocols on Resource Constrained Devices

Guido Moritz, Steffen Prüter, Dirk Timmermann
University of Rostock, Rostock 18051, Germany {guido.moritz, steffen.prueter, dirk.timmermann}
@uni-rostock.de

Frank Golatowski
Center for Life Science and Automation, Rostock
18119, Germany frank.golatowski@celisca.de

***Abstract* - Service-oriented architectures (SOA) become more and more important in connecting devices with each other. The main advantages of Service-oriented architectures are a higher abstraction level and interoperability of devices. In this area, Web services have become an important standard for communication between devices. However, this upcoming technology is only available on devices with sufficient resources. Therefore, embedded devices are often excluded from the deployment of Web services due to a lack of computing power and memory. Furthermore, embedded devices often require real-time capabilities for communication and process control. This paper presents a new table driven approach to handle real-time capable Web services communication, on embedded hardware through the Devices Profile for Web Services.**

## I. Introduction

THE usage of a standardized device-centric SOA is a possible way to fulfill interoperability requirements in future networked embedded systems. Technologies like UPnP (Universal Plug and Play), DPWS (Devices Profile for Web Services), REST (Representational state transfer) and Web services are used to realize a so called SODA (Service-oriented device architectures) [23]. While UPnP, DLNA and related technologies are established in networked home and small office environments, DPWS is widely used in the automation industry at device level [24] and it has been shown that they are also applicable for Enterprise integration [22, 26].

Besides the advantages of SODA, additional resources are required to host a necessary software stack. There are SODA toolkits available for resource-constrained devices like UPnP stacks or DPWS toolkits [WS4D, SOA4D]. However, additional effort is necessary for deeply embedded devices and for embedded real-time systems especially. Deeply embedded devices are small microcontrollers with only a few kB of memory and RAM (e.g. MSP430, ARM7). These devices cannot be applied to huge operating systems. But they are essential because as they combine price, low power properties, size and build-in hardware modules.

In this Paper a new approach is presented, which can be applied to deeply embedded devices and serve real-time and specification compliant DPWS requests.

## II. Web services in Device Controlling Systems

The World Wide Web Consortium (W3C) specifies the Web services standard [11]. UPnP is a popular specification in the home domain. Due to the lack of security mechanisms and the missing service proxy it is limited to small networks (see [2]). Web services are more important when using larger networks. This client-to-server interaction uses SOAP [10] for the transport layer and Extensible Markup Language (XML) for the data representation [1, 13]. On the other hand, the Web services protocols need much computing power and memory, in order to enable a device-to-device communication. Therefore, a consortium lead by Microsoft has defined DPWS [4]. DPWS uses several Web services protocols, while keeping aspect of resource constrained devices. In comparison to standard Web services, DPWS is able to discover devices at run time dynamically, without a global service registry (UDDI). The included WS-Eventing [5] specification also enables clients to subscribe for events on a device. The device sends a notification to the client, whenever an event occurs. State changes not have to be polled by the client. DPWS is integrated in the Microsoft operating system Vista. For many companies, this is the reason for developing new interfaces for their products based on these protocols.

In specific scenarios, communication proxies are necessary because of the low memory and computing power of deeply embedded devices. With the new implemented approach, Web services become also available on deeply embedded devices. Both deeply embedded devices and devices that are more powerful will be enabled to communicate and interact with each other. This substitutes the communication proxies.

Through linking the devices to a higher level of communication, devices no longer rely on specific transfer technologies like Ethernet. All devices in an ensemble are connected via services. This services based architecture is already used in upper layers. Services based communication becomes available on lower layers nearest to the physical tier. This allows a higher abstraction level of process structures. The first step to allow this is the creation of a SODA framework that fulfills the requirements of deeply embedded devices.

## III. Requirements for a Light Weight SODA

High-level communication on resource constrained embedded devices can result in an overall performance degradation. In a previous paper [6] we have presented different challenges which have to be met in order to realize DPWS communication with real-time characteristics.

This work was partly supported by the ITEA LOMS project.

Firstly, as a basis an underlying real-time operating system must exist, ensuring the scheduling of the different tasks in the right order and in specific time slots. Secondly, the physical network has to provide real-time characteristics. The major challenge in DPWS with respect to the underlying network, is the binding of DPWS and SOAP. SOAP is bound to the Hypertext Transfer Protocol (HTTP) for transmission. HTTP is bound to the Transmission Control Protocol (TCP) [8] (see Figure 1). The TCP-standard includes non-deterministic parts concerning a resend algorithm in case of an error. Furthermore, the Medium Access Control (MAC) to the physical tier has to grant access to the data channel for predictable time slots. For example, Ethernet cannot fulfill this requirement.

As shown in Figure 1, it is possible to use SOAP-over-UDP. But in accordance to the DPWS specification, a device must support at least one HTTP interface [4].

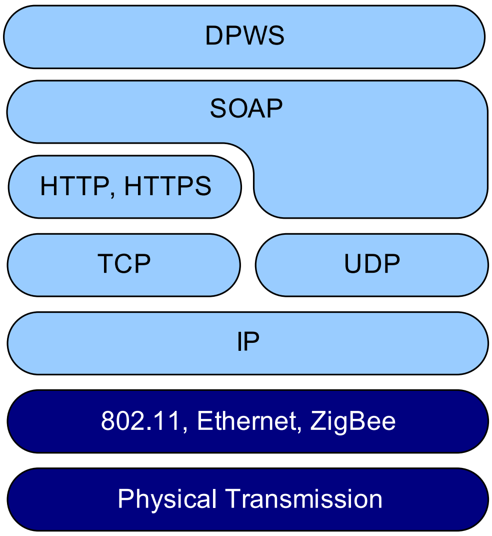


Figure 1. DPWS protocol stack

In Prüter et al. Xenomai [9] is used as operating system and RTNet [12] is used to grant network access with real-time characteristics. RTNet relies on the User Datagram Protocol (UDP) instead of TCP and uses Time Division Multiple Access (TDMA) for Medium Access Control (MAC). The usage of UDP demands SOAP-over-UDP at the same time. At least two interfaces have to be implemented: A non real-time, DPWS compliant HTTP/TCP interface and a real-time UDP interface. The disadvantage of using a special network stack including a special MAC, also implies building up a separate network. In this network, all participating notes have to conform to the MAC and used protocols.

For deeply embedded devices, various real-time operating systems exist. FreeRTOS[19] is a mini real-time kernel for miscellaneous hardware platforms like ARM7, ARM9, AVR, x86, MSP430 etc. Unfortunately, no useful real-time network stack and operating system combination is currently available for these kinds of deeply embedded devices. Therefore, this paper concentrates on the possibilities to provide real-time characteristics in the upper layers being on the top of TCP/IP.

The binding of DPWS and TCP through HTTP causes different challenges in granting real-time characteristic for DPWS communication and is still an ongoing work in our research group. It is not possible to reach deterministic characteristics without specific real-time operating systems and network stacks. A real-time operating system grants access to peripheries for predictable time slots and execution of tasks in the right order. The arising high level communication may not interfere with the real-time process controlling. The underlying real-time operating system takes care of correct thread management and correct scheduling of the real-time and non real-time tasks. Tasks on the controller, competing with the communication, are prioritized by the operating system.

In order to provide Web services on microcontrollers, different challenges have to be met. Figure 2 shows the single parts, which have to be realized.

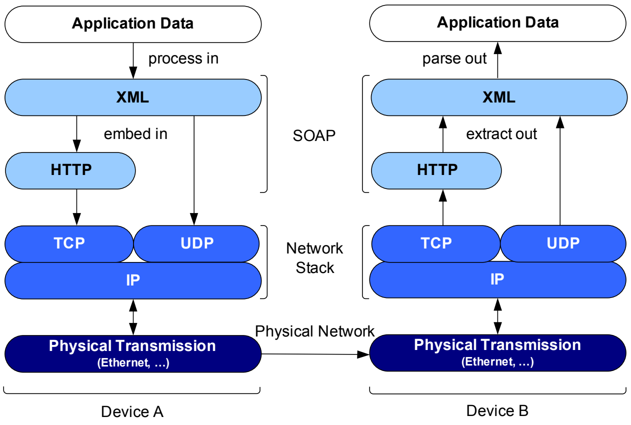


Figure 2. Modules to be implemented

### *A. Network Stack*

The network stack, responsible for the right addressing and the way of exchanging data, is the first module, which have to be realized. Dunkels has developed uIP and lwIP, two standard compliant TCP/IP stacks for 8 Bit controller architectures ([13, 14, 15]). uIP fulfills all minimum requirements for TCP/IP data transmissions. The major focuses are minimal code size, memory and computing power usage on the controller, without losing standard conformance. lwIP also fulfills non mandatory features of TCP/IP. Both implementations are designed to run on 8-bit architectures with and without an operating system. The differences between both stacks are shown in the following table.

DPWS utilizes WS-Discovery for automatic discovery of devices and is based on IP Multicast. Multicast applications use the connectionless and unreliable User Datagram Protocol (UDP) in order to achieve multicast communications. uIP is able to send UDP Multicast messages, but is not able to join multicast groups and receive multicast messages [15]. In contrast to uIP, the lwIP implementation supports all necessary UDP and Multicast features. The above mentioned FreeRTOS can use the lwIP stack for networking. This combines the advantages of a compatible, lightweight network stack and the usage of an embedded real-time operating system.

Table 1.
uIP vs. lwIP

| Feature | uIP | lwIP |
|---|---|---|
| IP and TCP checksums | X | X |
| IP fragment reassembly | X | |
| IP options | | X |
| Multiple Interfaces | | X |
| **UDP** | | **X** |
| Multiple TCP connections | X | X |
| TCP options | X | X |
| Variable TCP MSS | X | X |
| RTT estimation | X | X |
| TCP flow control | X | X |
| Sliding TCP window | | X |
| TCP congestion control | Not needed | X |
| Out-of-sequence TCP data | | X |
| TCP urgent data | X | X |
| Data buffered for rexmit | | X |

### *B. SOAP*

Upon the network stack, HTTP communication protocol is used. The payload is embedded in XML structures and sent via HTTP.

Because DPWS requires a small part of the HTTP functionality only, it is not necessary to implement a full functional HTTP stack. All DPWS messages are using the POST method of HTTP for delivering.

In contrast, the XML processing and parsing draws more attention. On deeply embedded devices, with only few kB of memory, the code size and the RAM usage have to be reduced. The WS-Discovery and WS-Metadata messages exceed the Maximum Transmission Unit (MTU) of most network technologies, including Ethernet. This supports the decision for lwIP in favour of uIP. The uIP implementation only uses one single global buffer for sending and receiving messages. The application first has to process the message in the buffer, before it can be overwritten [15]. In case of a complete XML message, the whole file has to be available before a useful parsing can be processed. Additional, computing power is restricted to resource constrained devices. With respect to the overall performance of the communication task, it is difficult to work through and parse the whole message as a nested XML file. Therefore, our research group has developed and implemented a new approach to handle HTTP and XML analysis. This new approach is described in the next section.

## IV. New Table Driven Approach

A complete implementation of SODA for deeply embedded systems, like wireless sensor network nodes with limited processing power and memory, is a significant challenge. All modules that are mentioned in section III like network stack, SOAP, HTTP and DPWS have to be implemented.

We have analyzed different setups with DPWS compliant implementations to identify which parts of DPWS could be omitted to reduce necessary resources. In most scenarios, only few types of messages have to be processed. After discovery and metadata exchange, the devices and their addresses are known and the services can be invoked. Only a few parts change within the exchanged messages. Major parts of the messages stay unchanged. Every time a service is called, almost the same message has to be parsed and almost the same message has to be build.

With all exchanged messages from the analysis of different scenarios tables are generate. The tables contain all appropriated incoming and outgoing messages. The new implemented table driven approach is able to answer every request with the right answer by referring to these tables. In section the dynamic changing parts of the different messages are shown.

This new table driven implementation is not based on SOAP and HTTP. Instead, we are using an approach basing on a simple string comparison of incoming messages in this new implementation. The messages are interpreted as simple text messages and not as SOAP Envelopes being embedded in HTTP. The relevance of the received strings from HTTP and SOAP protocols are unknown for the table driven device. The device is able to send specific response with the correctly adapted dynamic changing sections. The overhead for parsing and building always the same message is reduced by this approach. Thereby memory usage and computation time are decreased in comparison to a traditional implementation.

With respect to a real-time capable communication, the treatment of the messages as strings and not as specific protocols is useful. The parsing as a string is independent of the depth of the nesting of XML structures. The necessary time, to parse the message as a string, is predictable. XML Schema, which is required by DPWS, cannot fulfill these requirements.

## V. Mobile Robots as Test Bench

We verified our solution in a real world scenario. An external PC and an overhead camera control a team of five autonomous robots. The robots are coordinated via DPWS interfaces. The robots receive commands from a central server. The commands have to be executed in predicted timeslots to prevent collisions and enable accurate movement of the robots. The whole setup is shown in Figure 3.

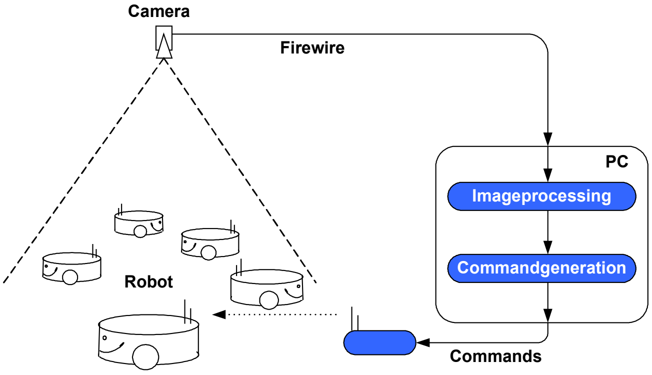


Figure 3. Robot Scenario

The team behavior of the robots is controlled by a central server which uses one or more cameras mounted above the ground. Image processing software on the PC extracts the position of all robots in the field. On the PC even the commands for the robots are calculated. These commands consist of global coordinates of the robot positions and the target positions. These commands are sent with a high transmission rate to the robots. The robots use global coordinates to update their own local and imprecise coordinate tracking. The robots need this global updates in regular periods, otherwise a correct controlling cannot be granted. These real-time requirements for controlling the robots with a parallel running communication system make the robot scenario an ideal test bench for our implementations.

### *A. Robot Hardware*

To control the robots we use two controller boards alternatively: an embedded Linux board and an ARM7 controller board. The embedded Linux board is the Cool Mote Master (CMM) from LiPPERT. It is equipped with an AMD Alchemy AU 1550 processor [17]. This board is designed as a gateway node for sensor networks. The CMM is already equipped with an 802.15.4 compliant transceiver. We have extended the board with additional Bluetooth and Wi-Fi (IEEE 802.11) interfaces [18]. Thereby, the board has three different radio technologies for networking beside Ethernet.

The ARM7 board is a SAM7-EX256 evaluation board from Olimex . This board is applied with an Atmel ARM7 controller with 256 kB memory and 64 kB RAM. The board already provides an Ethernet interface, which was used for testing. The controller is running with a clock rate of 55MHz. It is possible to schedule the lwIP stack and the implemented table driven device in different tasks with the help of FreeRTOS.

The implementations are evaluated on a standard PC and on these boards. An overview of used hardware is provided in Table 2 . The network devices are configured in a way, that all of them can handle IP traffic.

Table 2.
Used hardware for testing the new table driven approach

| | PC | CMM | SAM-7 |
|---|---|---|---|
| CPU | Intel Pentium4 | AlchemyAU 1550 | Atmel ARM7 |
| Clock | 3,4GHz | 500MHz | 55MHz |
| ROM | 500 GB | 512MB | 256 kB |
| RAM | 1024 MB | 256MB | 64 kB |
| Operating System | Linux (2.6.24/ Ubuntu) | Linux (2.6.17/ Debian) | FreeRTOS 5.0 |
| Network interfaces | Ethernet | Ethernet,<br>802.11g,<br>802.15.4,<br>Bluetooth | Ethernet |

## VI. Implementation

Our research group has implemented the WS4D-gSOAP toolkit[7]. This is a software solution, which includes a DPWS stack and software tools for creating of own Web services based on DPWS. This toolkit uses gSOAP [20] for SOAP functionalities and extends gSOAP with an implementation of the DPWS specification. This traditional implementation will be used as benchmark for the new table driven approach.

In the first step a service is created with the existing WS4D toolkit that provides all necessary commands for the robots in our mobile robot scenario. The external PC calls a hosted service on the robots. The service is called every time when new commands have to be send to the robots. The new commands are embedded in the request. The service answers with a response, including a performance parameter of the robot.

In the second step, the exchanged messages are analyzed according to the DPWS specification. All possible outgoing and incoming messages for the mobile robot scenario are generated.

In the third step, a completely new device is implemented. The structures and contents of the possible messages are deposited in the new implemented device as strings. This device does not support any dynamic SOAP or HTTP functionalities. The new table driven approach does not parse the whole incoming message as XML file. Every received message is analyzed with an elementary string compare. Thereby the type of the message is figured out. If the message type is known, the device answers with the right message. The answer is already deposited in the implemented device as a string also. In the answer, only parts required by the DPWS specification and the payload are changed.

During the implementation of the table driven device, we have taken care that system functions are not called in critical sections. For example, the memory management is provided by the task itself. The task allocates a pool of memory when it is started and then organizes the memory itself. Furthermore, the different threads for the network stack and the threads handling the messages are analyzed to be scheduled in the right order and with correct priorities.

### *A.Message Exchange*

Figure 4 gives an overview of exchanged messages in the mobile robot scenario. When starting the device, it announces itself with a *Hello SOAP Envelope*. Within this message only the wsa:MessageID and the wsd:XAddrs, the IP address, are dynamically and has to be adapted. Furthermore, the MessageNumber and the InstanceID has to be correct.

When a client was not started, as the device announces itself with a *Hello* , the client asks with a *Probe* for available devices. The answer is a *Probe Match* , where the wsa:RelatesTo has to fit to the wsa:MessageID of the *Probe* and the wsa:MessageID has to be dynamic. Here, also the MessageNumber and the InstanceID has to be incremented.

When the devices and their addresses are known, the client will ask for the hosted services on the device in the next step. Therefore, a *GetMetadata Device* is send to the hosting service, which is at least a service that announces the available hosted services. The *GetMetadata* message is the first one that is sent via HTTP. Within the HTTP header, the content length, the length of the message, and the IP address has to be adopted. The address only has to be changed, if it was not known at compile time. This applies to all IP addresses in the scenario. In the *GetMetadata Device* message, the wsa:To entry has to match to the address of the device, detected through the *Probe* . The device answers with a *GetMetadata Device Response* message. In this message the wsa:RelatesTo has to match the wsa:MessageID of *the GetMetadata Device*.

When the client knows available hosted services, the specific hosted service, that the client is looking for, is asked for the usage interface with a *GetMetadata Service* . The *GetMetadata Service Response* refers to the *GetMetadata Service* through the wsa:RelatesTo section.

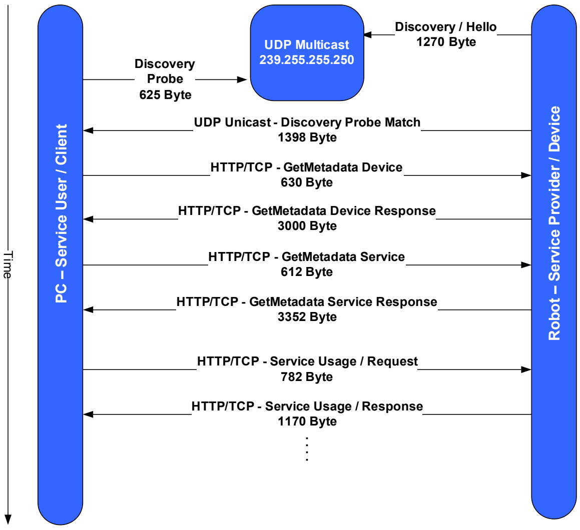

Time
PC – Service User / Client
Robot – Service Provider / Device
UDP Multicast
239.255.255.250
Discovery / Hello
1270 Byte
Discovery
Probe
625 Byte
UDP Unicast - Discovery Probe Match
1398 Byte
HTTP/TCP - GetMetadata Device
630 Byte
HTTP/TCP - GetMetadata Device Response
3000 Byte
HTTP/TCP - GetMetadata Service
612 Byte
HTTP/TCP - GetMetadata Service Response
3352 Byte
HTTP/TCP - Service Usage / Request
782 Byte
HTTP/TCP - Service Usage / Response
1170 Byte

Figure 4. Message Exchange[1]

After the metadata exchange is complete, the client knows how to interact with the specific service and the service usage starts. The client invokes the service with a message, where wsa:To and wsa:MessageID has to be correct. In the *Service Usage Request* , the coordinates of the mobile robot scenario are integrated. The service answers with the *Service Usage Response* . Therein, the reference to the request is given through the wsa:RelatesTo section. In our special mobile robot scenario, the response also contains the mrs:ProcessingTime section. In this section, the service informs the service user about the time, the application needs to process the new coordinates and is a performance parameter for the mobile robot.

An overview about the dynamic parts of the different messages is given in Table 3 .

The overall size for the exchanged messages is 12.839 Bytes. The overall number of Bytes that can change is 588. Only 4.6% of the overall exchanged bytes are dynamic in the mobile robot scenario.

## VII. Measurements/Comparison

In this section, the performances of the original WS4D toolkit and the new table driven approach are compared.

Table 3. Overview Exchanged Messages

| MessageType | Changingparts | Dynamic Bytes |
|---|---|---|
| Hello | wsa:MessageID | 36 |
| | wsd:XAddrs(IP) | max.17[2,3] |
| | wsd:AppSequenceMessageNumber | approx.2 |
| | wsd:AppSequence InstanceId | 10 |
| Probe | wsa:MessageID | 36 |
| ProbeMatch | wsa:MessageID | 36 |
| | wsa:RelatesTo | 36 |
| | wsd:AppSequenceMessageNumber | approx.2 |
| | wsd:AppSequence InstanceId | 10 |
| GetMetada Device | HTTP content-length | max.5 |
| | HTTP host | max.17[2,3] |
| | wsa:MessageID | 36 |
| | wsa:To | 36 |
| GetMetadataDeviceResponse | HTTP content-length | max.5 |
| | wsa:RelatesTo | 36 |
| | wsa:Address | max.17[2,3] |
| GetMetadata Service | HTTP content-length | max.5 |
| | HTTP host | max.17[2,3] |
| | wsa:MessageID | 36 |
| | wsa:To | max.17[2,3] |
| GetMetadata Service Response | HTTP content-length | max.5 |
| | wsa:RelatesTo | 36 |
| Service Usage Request | HTTP content-length | max.5 |
| | HTTP host | max.17[2,3] |
| | wsa:MessageID | 36 |
| | wsa:To | max.17[2,3] |
| | mrs:Position[4] | 16 |
| Service Usage Response | HTTP content-length | max.5 |
| | wsa:RelatesTo | 36 |
| | mrs:ProcessingTime[4] | 3 |

### *A. Devices Sizes*

The WS4D toolkit implementation of the DPWS device needs 794 kB of disk space when compiled for Linux on a x86 architecture. The table driven device implementation has a 16 kB footprint when compiled for a standard x86 PC running with Linux. The table driven device does not contain the independent lwIP stack in this x86 implementation. Both implementations for a x86 PC running with Linux are using the BSD Socket API to handle the network traffic. The same implementation of the new table driven approach ported to the SAM7-EX256 board running with FreeRTOS 5.0 has a 13 kB footprint without network interface. As network stack the independent lwIP stack in Version 1.3 is applied to the board. Therefore, the stack was ported to FreeRTOS 5.0.

[1] number of Bytes may vary because of different IP addresses and payload e.g.

[2] depends on IP Address and Port number

[3] if not known at compile time

[42] depends on IP Address and Port number

[3] if not known at compile time

Payload

The required disk space for the different parts on the SAM7 board is shown in Table 4. The overall memory being used on the board, including FreeRTOS, lwIP and the device needs 146 kB.

Table 4.
Footprint of SAM7 Implementation with FreerRTOS and lwIP

| Module | Footprint |
|---|---|
| static DPWS device | 13 kB |
| lwIP 1.3 | 77 kB |
| FreeRTOS including Debug Tasks | 56 kB |

### B. Time Responds

Also some timing measurements have been done in order to have an objective comparison for the new static approach. Therefore, the overall time was measured that is required from sending the message to receiving the response on the client side. Through this method the overall performance and the maximum number of requests per second can be determined which can be served.

These measurements are done for a standard PC and the SAM7 board. On both devices a 100 MB/s Ethernet interface is applied, which has been used for the measurements. On the SAM7 boards, an independent thread simulated an additional CPU load. This CPU load thread was scheduled with different priorities. As requesting client a standard PC (2x3,5 GHz with 1 GB RAM) was used in all cases.

The following Table 5 shows the times measured for the different implementations of DPWS server/device. The values are the average over 1000 requests, send directly successive.

Table 5.
Respond time and processed requests per second

| Device System | Response time | Requests per sec |
|---|---|---|
| Standard PC<br>WS4D toolkit | 1,05 ms | 952 |
| Standard PC<br>Table Driven DPWS | 0,9 ms | 1111 |
| SAM7-EX256<br>Table Driven DPWS | 18,6 ms | 53 |
| SAM7-EX256<br>Table Driven DPWS<br>CPU load thread lower priority then lwIP and DPWS tasks | 18,6 ms | 53 |
| SAM7-EX256<br>Table Driven DPWS<br>CPU load thread same priority like lwIP and DPWS tasks | 30,2 ms | 33 |

On the PC, the new implemented approach provides a faster overall processing. The time responds on the real-time operating system of the SAM7 board, depend on given priorities for the different competing tasks. As long as the CPU load task has a lower priority than the DPWS and the lwIP tasks, no effect to the average times could be measured.

## VIII. Conclusion

The new table driven approach allows the usage of Web services on deeply embedded devices. Furthermore, the implemented services can grant real-time capabilities. Thus, the deeply embedded devices can be integrated in enterprise service structures. The created service interfaces can be reused in different application. The connectivity between such large numbers of embedded devices normally needs proxy concepts with static structures. Now, these proxies are no longer required. The devices can be directly accessed by a high level process logic. Furthermore, the validation and certification become cheaper because of the slim implementation and reusability of the interfaces.

The measurements show that the size of a device can be reduced by the factor of about 50. At the same time, the time responds can be improved. Through the implementation in different threads, the time responds of the new implemented static approach is independent from other competing tasks. However, this assumes an underlying real-time operating system.

The optimization of the footprint and dynamic memory usage are a main focuses for the future work. Future work will also research on a completely specification compliant implementation. Therefore, the specification has to be analyzed in detail and all possible messages, even for error cases, have to be discovered and integrated into the static device.

## References

[1] W. Dostal, M. Jeckle, I. Melzer, and B. Zengler, *Serviceorientierte Architekturen mit Web Services*. Elsevier, 2005.

[2] Elmar Zeeb, Andreas Bobek, Hendrik Bohn, Frank Golatowski, "Service-Oriented Architectures for Embedded Systems Using Devices Profile for Web Services," in *2nd International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 2007)*, pages 956-963, Niagara Falls, Ontario, Canada, May 21-23, 2007.

[3] Marco Sgroi, Adam Wolisz, Alberto Sangiovanni-Vincentelli and Jan M. Rabaey, "A Service-Based Universal Application Interface for Ad-hoc Wireless Sensor Networks (Draft)," *Unpublished article*, November 2003.

[4] Microsoft Corporation, *DPWS Specification*, Technical Report, Microsoft Corporation, http://specs.xmlsoap.org/ws/2006/02/devprof, 2006.

[5] World Wide Web Consortium, *Web Services Eventing (WS-Eventing) Submission*, Technical report, http://www.w3.org/Submission/WS-Eventing/, March 2006.

[6] Steffen Prüter, Guido Moritz, Elmar Zeeb, Ralf Salomon, Frank Golatowski, Dirk Timmermann, "Applicability of Web Service Technologies to Reach Real Time Capabilities," *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC),* Orlando, Florida, USA, pages 229-233, May 2008.

[7] University of Rostock. *DPWS-Stack WS4D*, Technical Report, University of Rostock, http://ws4d.org, 2007.

[8] Internet Engineering Task Force, *Hypertext Transfer Protocol--HTTP/1.1*, Technical Report, http://tools.ietf.org/html/rfc2616, 1999.

[9] Philippe Gerum, "Xenomai - Implementing a RTOS emulation framework on GNU/Linux," Whitepaper, 2004.

[10] World Wide Web Consortium, *Simple Object Access Protocol Specification,* Technical report, World Wide Web Consortium, http://www.w3.org/TR/soap/, 2008.

[11] World Wide Web Consortium, *Web Service Architecture Specification*, Technical report, World Wide Web Consortium, http://www.w3.org/TR/ws-arch/, 2007.

[12] Kiszka, J. and Wagner, B. and Zhang, Y. and Broenink, J.F., “RTnet - A flexible Hard Real-Time Networking Framework,” *10th IEEE International Conference on Emerging Technologies and Factory Automation Volume 1*, Catania, Italy, page 8, September 2005.

[13] Adam Dunkels, “Full TCP/IP for 8-Bit Architectures ,” *International Conference On Mobile Systems, Applications And Services* , San Francisco, California, pages 85-98, 2003.

[14] Adam Dunkels, *uIP* , Technical report, http://www.sics.se/~adam/uip/index.php, 2007.

[15] Adam Dunkels, *lwIP - A Lightweight TCP/IP stack* , Technical Report, http://savannah.nongnu.org/projects/lwip/, 2008.

[16] Adam Dunkels, *The uIP Embedded TCP/IP Stack*, The uIP 1.0 Reference Manual, Technical report, 2006.

[17] LiPPERT: PC solutions for rugged industrial applications, *Cool Mote Master Board*, Technical report, http://www.lippert-at.com/, 2008.

[18] Thorsten Schulz, *Evaluierung verschiedener Prozessorlösungen für RoboCup Roboter* , Technical report, 2006 .

[19] *FreeRTOS - The Standard Solution For Small Embedded Systems*, http://www.freertos.org, 2008.

[20] Robert A. van Engelen, Kyle A. Gallivany, “The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks,” *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID)* , Washington, DC, USA, page 128, May 2002.

[21] Olimex, *SAM7-EX256 Evaluation Board*, Technical report, http://www.olimex.com/dev, 2008.

[22] S. Karnouskos, O. Baecker, L. Moreira, S. de Souza, P. Spieß, “Integration of SOA-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure,” *Emerging Technologies & Factory Automation (ETFA)*, Patras, Greece, pages 293-300, Sept. 2007.

[23] Scott de Deugd, Randy Carroll, Kevin E. Kelly, Bill Millett, and Jeffrey Ricker, "SODA: Service-Oriented Device Architecture," *IEEE Pervasive Computing* , vol. 5, no. 3, pages 94-C3, 2006.

[24] H. Bohn, A. Bobek, and F. Golatowski, “SIRENA - Service Infrastructure for Realtime Embedded Networked Devices: A service oriented framework for different domains *”, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL’06)* , page 43, Washington, DC, USA, 2006.

[25] Elmar Zeeb, Steffen Prueter, Frank Golatow ski, Frank Berger, "A context aware service-oriented maintenance system for the B2B sector," *3rd International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 2008)*, Ginowan, Okinawa, Japan, pages 1381-1386, March 26, 2008.

[26] Intel, *UPnP - Technology Overview* , Technical report, http://www.intel.com/cd/ids/developer/asmona/eng/downloads/upnp/overview/index.htm , 2008.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 703 – 709

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Task jitter measurement under RTLinux and RTX operating systems, comparison of RTLinux and RTX operating environments

Pavel Moryc
Technical University of Ostrava
Faculty of Electrical Engineering
and Computer Science
Department of Measurement and
Control
Centre for Applied Cybernetics
Ostrava, Czech Republic
Email:
pavel.moryc@mittalsteel.com

Jindřich Černohorský
Technical University of Ostrava
Faculty of Electrical Engineering
and Computer Science
Department of Measurement and
Control
Centre for Applied Cybernetics
Ostrava, Czech Republic
Email:
jindrich.cernohorsky@vsb.cz

*Abstract* — **This paper compares task jitter measurement performed under RTLinux and RTX hard real-time operating systems. Both the operating environments represent Hardware Abstraction Layer extensions to general-purpose operating systems, and make possible to create a real-time system according to the POSIX 1003.13 PSE 54 profile (multipurpose system). The paper is focused on discussion of experimental results, obtained on PC hardware, and their interpretation.**

## I. Introduction: Applicability of general-purpose operating systems in control systems

REAL-TIME task is a task, which meets prescribed deadline. Control system is a computer system, which physically realizes real-time tasks. The physical realization of the real-time task is not ideal, but it shows latencies and jitters.

Operating system and hardware support the real-time tasks with standardized means, which have non-zero and variable overhead. Kernel latency is defined as a delay between scheduled and actual execution of a task (e.g. between scheduled and actual instance starting time) [5]. The kernel latency is not stable, but it shows a jitter. Jitter is defined as a variable deviation from ideal timing event. The causes of jitter encompass code branching into paths with different execution times, variable delays implied whenever variable amount of code or data is stored in or read from caches and buffers, as well as noise and electromagnetic interference.

The substance of jitter can be seen in a need for parallel processing, which in turn stems from inevitable conflict between predictability and unpredictability. The control system has to be predictable and deterministic as much as possible. Control systems are intended to communicate with technology, as well as with people managing the technology (at least an emergency stop pushbutton is provided). Seen from this viewpoint, control system shall be appropriately flexible. As the result, a control system design represents a compromise between stability and adaptability.

It is possible to design a control system on two computers, one designed to meet technology demands, and the other designed to meet the user interface demands, but there is also the possibility to realize such system on one computer only. Computer system of this scope is standardized in the POSIX 1003.13 standard, as the PSE 54 profile (multipurpose system). PSE 54 - sized solutions are often preferred for their challenging possibility to use a broad range of standardized and low-cost hardware components, primarily intended for general-purpose computers, but on the other hand, they can involve lack of predictability typical for general-purpose (non real-time) systems.

Traditional general-purpose kernel provides full range of API services specified in the POSIX 1003.1 standard, and because of that, it cannot guarantee appropriately deterministic behavior, required in most real-time and technology control applications. Basic approaches to make such architecture more real-time one include

- low-latency kernel,
- preemptible kernel,
- hardware abstraction layer (HAL).

Low-latency kernel represents a traditional approach. The low-latency kernel is monolithic (i.e. it cannot be preempted by task), but its design minimizes latencies and jitters of the kernel API services typically used in real-time applications.

Preemptible kernel can be preempted by task. Preemptivity by task means, that a task can preempt the kernel just servicing another task, and enter the kernel instead. This type of premptivity is called reentrancy. However, at least some parts of a kernel (scheduling, interrupt service mechanism) cannot be made reentrant. Moreover, the idea of a preemptible kernel itself does not imply, that the kernel is deterministically preemptible, i.e. its jitters are acceptably stable.

Last but not least, hardware abstraction layers can be applied. A hardware abstraction layer receives timer interrupt and sends a virtual (software) interrupt to the general-purpose operating system kernel, thus providing a virtual (slower) clock for the general-purpose operating system. This can be seen as a cycle stealing. In the free time, real-time tasks can be run. Obviously, latencies and jitters of the HAL layer must be kept low enough by design.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

## II. RTLinux and RTX basics

Both the RTLinux and RTX represent real-time Hardware Abstraction Layers. RTLinux is the product of the FSMLabs, Inc., designed for the Linux operating environment, while RTX is the product of the Citrix Systems Inc. (Ardence), and is designed to run under the Windows operating system.

### *A. RTLinux*

RTLinux microkernel (fig. 1a) implements a Hardware Abstraction Layer inserted between hardware and Linux kernel. Both the RTLinux microkernel and the Linux kernel have to communicate mutually, thus it is necessary to make certain modifications into the Linux kernel. The modifications include

- modification of macros for disabling and enabling interrupts in order to use internal HAL signals (software interrupts) instead of using cli and sti assembler instructions,
- modification of interrupt handlers, in order to use signals instead of direct Interrupt Controller access,
- modification of device drivers in order to prevent them from using sti/cli assembler instructions directly as well.

Within a RTLinux thread, time is measured with resolution that depends on hardware. On the Pentium 4 platform, the time resolution is equal to 32 ns.

[1] and [4] discuss the overall architecture more deeply. Basic resources used for interprocess synchronization and IPC communication in the RTLinux/Linux environment are semaphores, mutexes, shared memory pools, and real-time data pipes, which are more complex structures combining buffers and mutexes. Moreover, many A/D cards are controlled by direct access to registers, thus direct I/O access is an important feature.

### *B. RTX*

RTX microkernel is similar to RTLinux kernel in functionality, but different in realization (fig. 1b). It supports real-time tasks, which are called RTSS threads. The Windows kernel is a proprietary solution, and its source code is not freely available to the public. Fortunately, two possible access paths to its modification exist. The first customizable part is the Windows HAL, and the second one is a device driver [2]. Basically, it is necessary to modify the Windows HAL for three purposes:

- to add interrupt isolation between Windows kernel and RTSS threads,
- to implement high-speed clocks and timers,
- to implement shutdown handlers.

The two interconnection points between Windows kernel and RTX microkernel mentioned above make possible to realize connection between RTX microkernel and Windows kernel, providing the same functionality as the interface between RTLinux microkernel and the Linux kernel (fig. 1b), [2, fig. 1]. The communication interface between Windows and RTX kernels implements a low-latency client-server mechanism, which includes both buffers and Service Request Interrupts (SRI) [2].

Due to the communication interface, subset of Windows API services is callable from within a RTSS thread. It includes APIs for storing data to file, thus real-time pipes are not available in the API services set. However, we can reasonably suppose, that similar IPC mechanisms are necessary to provide similar functionality, no matter whether they are hidden for the programmer. Some of Windows APIs available from RTSS threads are listed as non-deterministic, i.e. they can cause significant jitter when called from a RTSS thread. High-speed (and high resolution) clocks are needed for real-time precise timer realization. Within a RTSS thread, time is measured with 100 ns step and resolution.

Shutdown handler is a mechanism delivering more robustness to the real-time RTSS subsystem when the Windows subsystem is crashed or regularly shut down.

It can be summarized, that following differences from RTLinux exist:

- no real-time pipes or their equivalents are available in the API service set,
- it is possible to call a subset of Windows API services directly from the RTX (RTSS) real-time task,
- time is measured with 100 ns resolution,
- interrupts to the Windows kernel are masked while the RTX (RTSS) real-time task runs,
- a real-time interrupt routine has two mandatory parts, which can be used as upper and bottom ISR part,
- it is possible to implement a shutdown handler as the last resort resource.

## III. Applied measurement method

The measurement method applied is described in [3] and more deeply in [4]. Based on RTLinux resource analysis, following important RTLinux characteristics have been identified:

- precision of scheduler (measured as task starting time jitter),
- interrupt latency time,
- execution time of typically used API services, e.g.
  - pipe write and read operations,
  - shared memory write and read operations,
  - thread switching time
- I/O port read and write access time.

The I/O access is also included, because it characterizes hardware, and presents the basic method of communication with both sensors and actuators.

A generalized application has been written, which uses the above-mentioned RTLinux key resources. The application is called RT-golem, and its design is described in [3] and [4].

As RTLinux and RTX operating environments are functionally similar and their key resources are merely the same, it can be supposed, that similar design can be used for jitter measurement under the RTX operating environment too.

The RT-golem is written in C language, thus it should be portable. But, both the environments contain non-portable extensions, and as a result, the design had to be partially re-written. The RT-golem re-written for the RTX environment is called Win-golem.

The measurement method is realized in measurement architecture. The measurement architecture includes both software (measurement and workload tasks and operating system), which all is a mere design abstraction, and hardware, which presents its physical realization. Nonetheless, we should note, that the border between software and hardware design is rather fuzzy, and many non-trivial resources formerly realized in software design are recently applied in hardware design too.

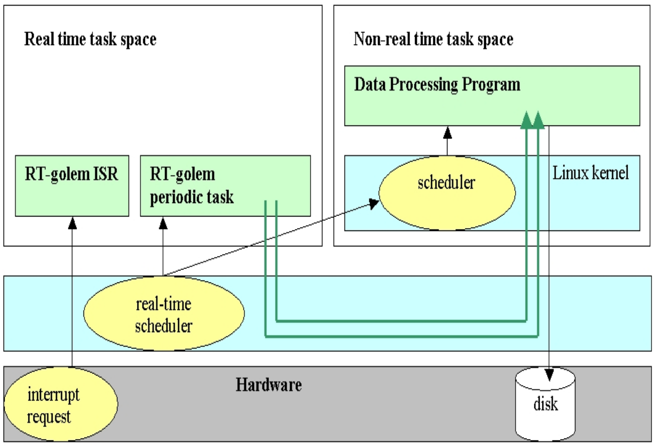


Fig. 1a RT-golem architecture

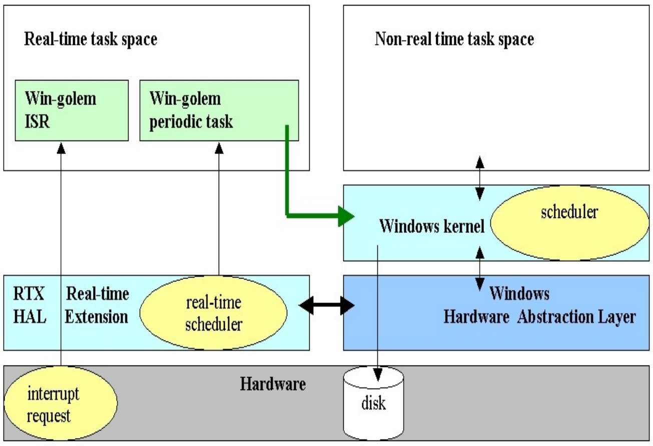


Fig. 1b Win-golem architecture

## IV. Experimental setup

Series of measurements have been performed. It has been measured on different hardware (PC Dell, PC no name), under different operating environments (RTLinux Free v. 3.1, RTX v. 8) and under different workload (basic workload only, basic workload and additional workload). Experimental setup configuration chart is given in Figure 1, while hardware configurations are presented in Table 1.

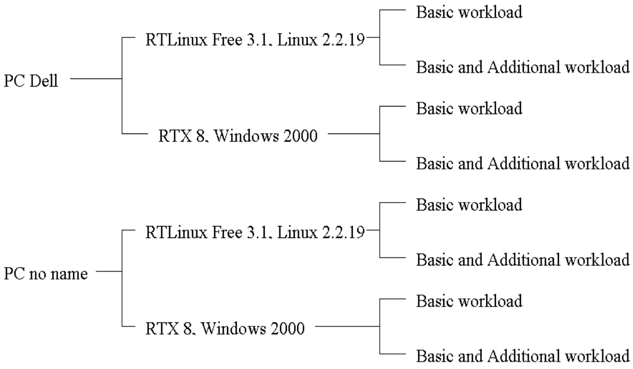

PC Dell
RTLinux Free 3.1, Linux 2.2.19
Basic workload
Basic and Additional workload
RTX 8, Windows 2000
Basic workload
Basic and Additional workload
PC no name
RTLinux Free 3.1, Linux 2.2.19
Basic workload
Basic and Additional workload
RTX 8, Windows 2000
Basic workload
Basic and Additional workload

Fig. 2 Experimental setup configuration chart

Basic workload means a workload caused by the operating system kernel (kernel overhead), daemons normally needed and running, and the measurement task. Additional workload is presented with a shell script copying short files (bash or command.com) in a loop.

**Table 1.** Test system details

**PC Dell GX 280**

| | |
|---|---|
| CPU | Intel P4 3.0 GHz, 1 MB L2 cache |
| RAM | 1024 MB |
| HDD | SAMSUNG SV0842D, SATA, 75GB |
| | WDC WD800JD-75JNC0, 8 GB, ATA-66 |

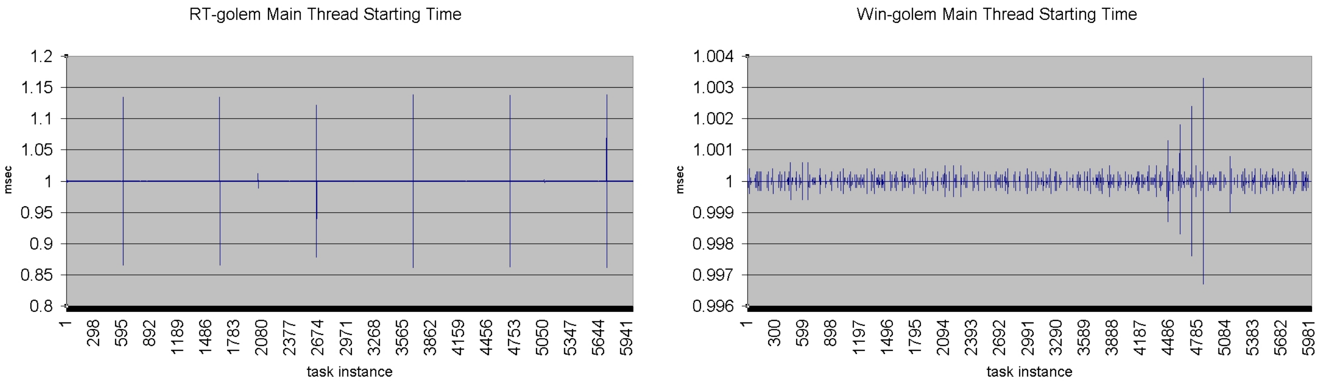


Fig. 3. RT-golem and Win-golem results comparison: Periodic Task Starting Time, PC Dell, basic workload only

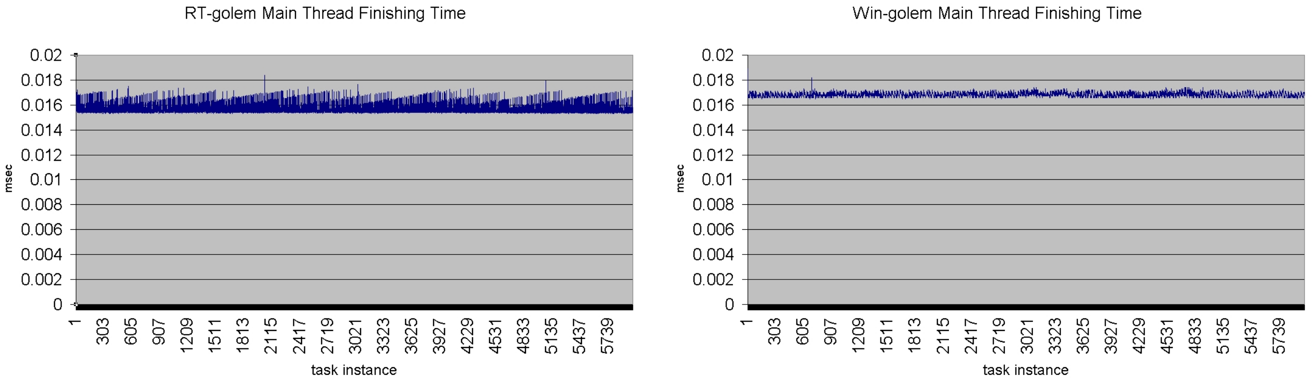


Fig. 4. RT-golem and Win-golem results comparison: Periodic Task Finishing Time, PC Dell, basic workload only

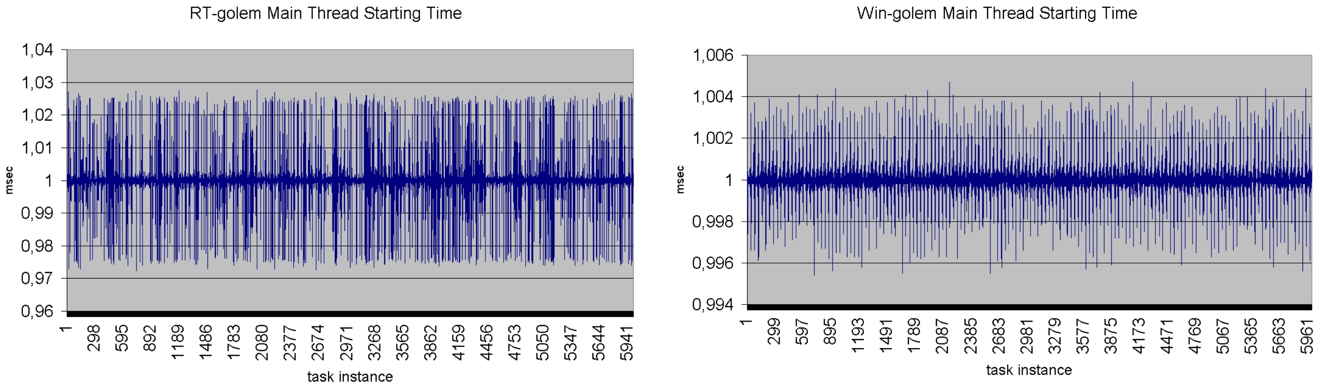


Fig. 5. RT-golem and Win-golem results comparison: Periodic Task Starting Time, PC Dell, basic and additional workload (copying files)

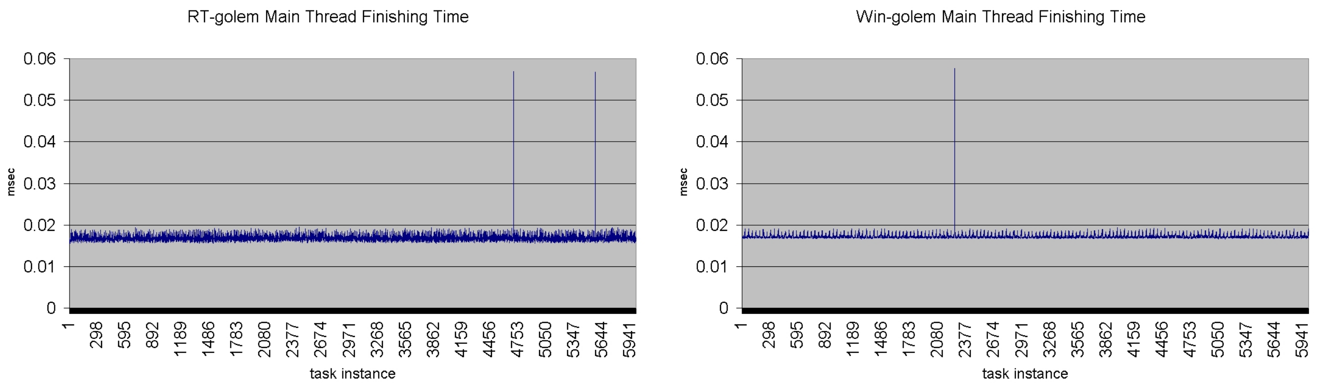
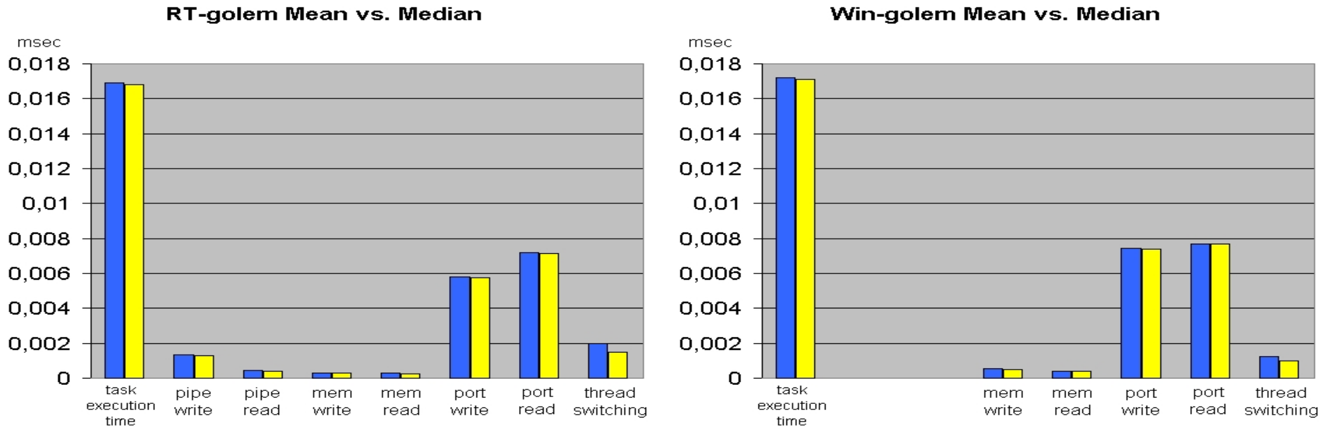


Fig. 6. RT-golem and Win-golem results comparison: Periodic Task Finishing Time, PC Dell, basic and additional workload (copying files)



blue: mean
yellow: median

Fig. 7. RT-golem and Win-golem results comparison: Execution Time Means vs. Medians, PC Dell, basic and additional workload (copying files)

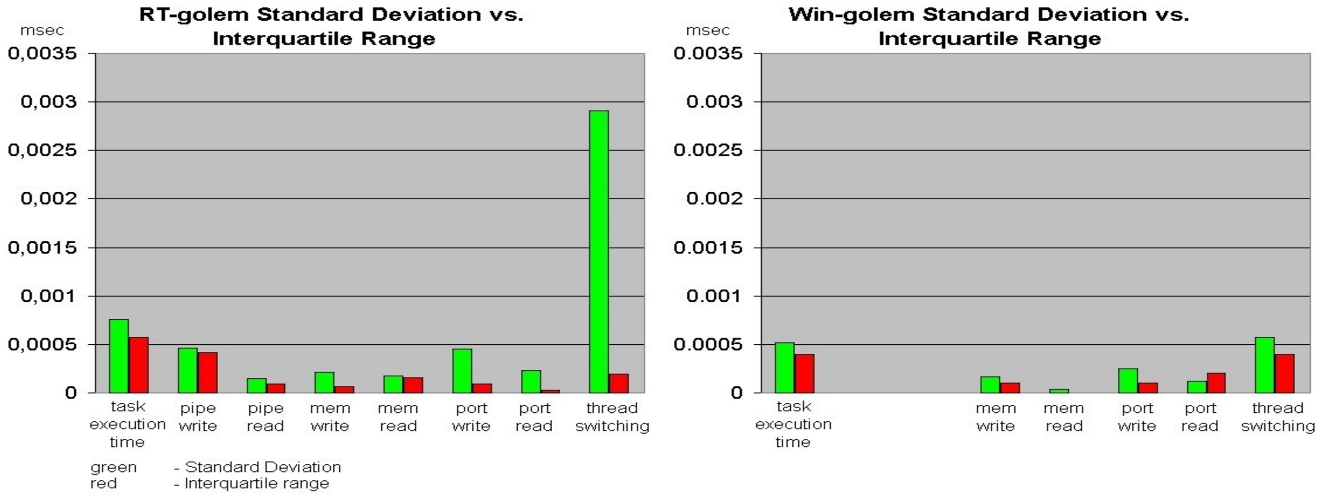


Fig. 8. RT-golem and Win-golem results comparison: Execution Time Standard Deviations vs. Interquartile Ranges, PC Dell, basic and additional workload (copying files)

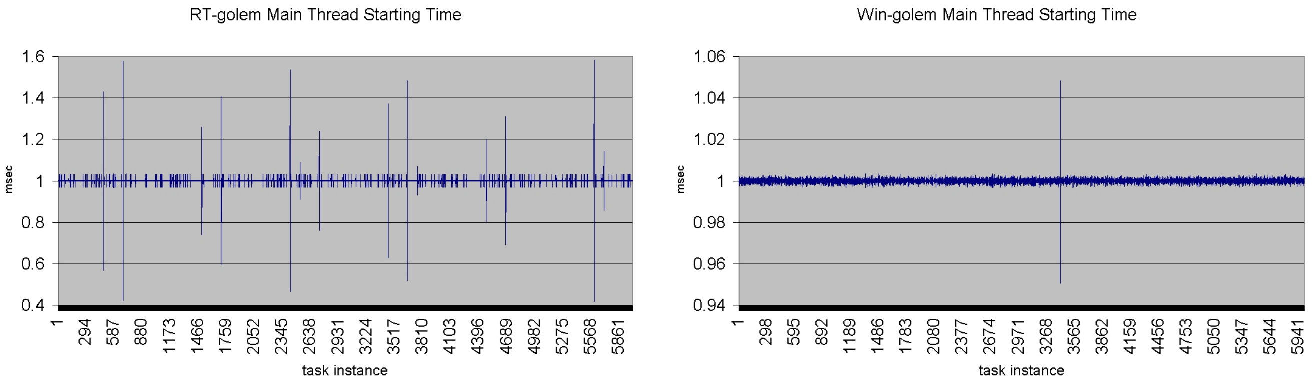


Fig. 9. RT-golem and Win-golem results comparison: Periodic Task Starting Time, PC no name, basic and additional workload (copying files)

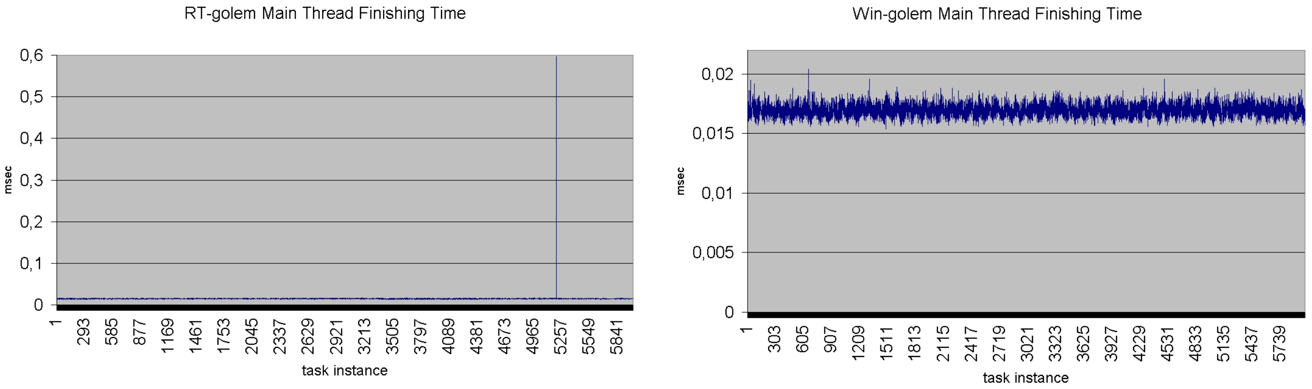


Fig. 10. RT-golem and Win-golem results comparison: Periodic Task Finishing Time, PC no name, basic and additional workload (copying files)

**TABLE 1.** (CONTINUED) TEST SYSTEM DETAILS

**PC NO NAME**

| | |
|---|---|
| CPU | Intel P4 2.4GHz, 32 K of L1 cache |
| mainboard | MSI 865 PE Neo2-P |
| RAM | 256 MB |
| HDD | Seagate Barracuda ST380011A 80 GB ATA-100 |
| | Maxtor WDC WD100EB-00BHF0 10 GB ATA-66 |

## V. EXPERIMENTAL RESULTS

Representative selection of results is presented further. The series of graphs, presented in fig. 3 through 6, and in figures 9 and 10, show the task instance starting (or finishing) times comparison. These graphs are completed with statistical data evaluation graphs (fig. 7 and 8), which show mean vs. median comparison and standard deviation vs. interquartile range comparison on the PC Dell platform.

The task instance starting time is calculated from the previous task instance starting time (as in [3]). This means, the starting time delay impacts two adjacent values. First, the difference between the correct and delayed instance is longer, which causes the spike up on the graph, and then, the difference between the delayed and next correct instance is shorter, which causes the spike down. If both spikes are symmetrical, the second value is okay. Finishing time is calculated from task instance starting time [3]. Spikes on the relative starting time graphs below oscillate around 1 msec, because they show scheduling jitter, i.e. a difference of the actual relative starting time from the nominal value, which is 1 msec.

## VI. CONCLUSION

It can be concluded from the presented experimental results, that the measured task starting time jitter (kernel jitter, [5]) is significantly lesser on the RTX-based measurement architectures, than on the RTLinux Free-based architectures (fig. 3, 5, 9). The median of task finishing time is approximately the same within the applied range of test architectures and workloads. Under RTLinux Free, the port write time median value is ca. 15% less than the port read time median value, but under RTX both medians are the same (fig. 7). As with the task starting time, task finishing time shows significantly lesser jitter on the RTX-based architectures (fig. 4, 6, 8, 10).

Using the RTLinux Free operating system, it has been observed (on graphs presented above and in [2], [3], and [4]), that most of the jitter instances are near the best-case values, but sometimes significantly higher spikes occur. These spikes can form typical patterns (measurement of task relative starting time, left graphs on fig. 3, 4, 9), or can be observed randomly (left graphs on fig. 6, 7, 10), but in any case their amplitude is typical for the underlying hardware. However, with the RTX and Windows system, the spikes are significantly less (fig. 3, right graph), or none at all.

It can be supposed, that the significant worst-case jitter spikes observed in experiments with RTLinux Free are caused by cache writing and flushing effects (in accordance with [2]). As the measurements are performed on the top of the hardware and software stack, and the virtual resources presented to the measurement task by the operating system API services are quite distant to the resources presented to the operating system by the hardware (more precisely, by the part of the operating system realized in hardware), it is not possible to validate such hypothesis by methods described here. However, the absence of this phenomenon on both test hardware with RTX operating system can imply, that the RTX microkernel prevents the hardware from flushing the cache freely. Moreover, some further tracks can be given. [2] notes video drivers as most cache demanding part of Windows operating system, and in the RTX platform evaluation kit, video is used as a workload. Video is a real-time task as well as a RTX microkernel task. Thus, the conflict between video and RTX microkernel can be seen as the conflict between two real-time cache-demanding tasks, which can lead to swapping the RTX code out of the cache.

Unfortunately, the RTX microkernel source code is not freely available, and it is not possible to verify the tracks given above with the code analysis. However, the measurement results as well as the tracks given above can suggest, that the mechanism of locking the real-time code in the hardware cache is worthy to be studied and implemented.

## ACKNOWLEDGMENT

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under Project TM 0567.

## References

[1] FSM Labs Inc., *"Getting Started with RT Linux"*, 2001.

[2] M. Cherepov et. al.: *"Hard Real-Time with Ardence RTX on Microsoft Windows XP and Windows XP Embedded"*, www.ardence.com, 2002.

[3] P. Moryc, J. Černohorský: *"Task jitter measurement under RTLinux operating system"*, in: Proceedings of the International Multiconference on Computer Science and Information Technology, ISSN 189-7094, pp. 849 to 858, 2007.

[4] P. Moryc: *"Měření procesů reálného času v operačním systému RTLinux"*, Doctoral Thesis, Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, 2007.

[5] I. Ripoll et al.: *"WP1: RTOS State of the Art Analysis: Deliverable D1.1: RTOS Analysis"*, 2002.

Proceedings of the International Multiconference on Computer Science and Information Technology pp. 711–718

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Multilevel Localization for Mobile Sensor Network Platforms

Jae Young Park, Ha Yoon Song
Department of Computer Engineering, Hongik University, Seoul, Korea
pjyooungs@hotmail.com, hayoon@wow.hongik.ac.kr

***Abstract*—For a set of Mobile Sensor Network, a precise localization is required in order to maximize the utilization of Mobile Sensor Network. As well, mobile robots also need a precise localization mechanism for the same reason. In this paper, we showed a combination of various localization mechanisms. Localization can be classified in three big categories: long distance localization with low accuracy, medium distance localization with medium accuracy, and short distance localization with high accuracy. In order to present localization methods, traditional map building technologies such as grid maps or topological maps can be used. We implemented mobile sensor vehicles and composed mobile sensor network with them. Each mobile sensor vehicles act as a mobile sensor node with the facilities such as autonomous driving, obstacle detection and avoidance, map building, communication via wireless network, image processing and extensibility of multiple heterogeneous sensors. For localization, each mobile sensor vehicle has abilities of the location awareness by mobility trajectory based localization, RSSI based localization and computer vision based localization. With this set of mobile sensor network, we have the possibility to demonstrate various localization mechanisms and their effectiveness. In this paper, the preliminary result of sensor mobility trail based localization and RSSI based localization will be presented.**

## I. Introduction

THE researches on Mobile Sensor Network (MSN) have been plenty worldwide. For MSN, there could be a lot of valuable application with attached sensors as well as capabilities such as locomotion, environmental information sensing, dead-reckoning, and so on. For such applications, usual requirements have been acknowledged with localization of each sensor node and formation of the whole sensor network. In this research, we are going to discuss about a Mobile Sensor Vehicle (MSV) which can compose MSN. We will discuss a construction of MSN as well as required functionalities of each MSN.

This paper is organized as follows. In section II we will discuss localization method that have been researched. The following section III we will analyze the requirement for MSV, the hardware design of MSV, and equipments for localization, and we will discuss software capabilities of MSV software and will show software components to fully control our MSV including software for MSN itself, monitoring program, map building features, and other related topics. In section IV, our approach and methodology for mobility trajectory based localization for medium distance localization will be discussed. Section V we will demonstrate one of the localization feature of our MSV. RSSI (Radio Signal Strength Identification) based localization will be presented based on 802.11 devices with software modification. We will show the merits and demerits of RSSI based localization. Finally section VI will conclude this paper with possible future research topics.

## II. Related Work

There have been a lot of researches regarding mobile sensor localization. In this section, we will discuss past researches concentrating RSSI based localization, and dead-reckoning based localization. This works are not restricted on mobile sensors only but also related to robot technology.

### A. RSSI based Localization

Radio Signal Strength Identification is one of the known solutions for distance measure. It requires wireless network device for mobile sensors and extra features.

We use 802. 11 network devices which have wide popularity. In addition we need communication between mobile sensors thus 802.11 networking devices are popular solutions for us. RSSI features for 802.11 device networks are required features in order to implement physical layer of CSMA/CA networking [12]. However RSSI based distance measure is very prone to radio signal attenuation and thus has low accuracy. And it has some restriction that once it is a data transfer modes, it cannot switched to API mode instantly. It implies the restricted realtimness for RSSI based localization. There are no defined standards for 802.11 RSSI and manufactures of 802.11 device usually provide their arbitrary method [11]. This is another restriction.

In this paper, we will demonstrate our MSV successfully does long distance, low accurate localization only with commercial 802.11 devices and networking software embedded on MSVs.

### B. Computer Vision Based Approach

There are very few researches on localizations by use of computer vision technology. There have been the previous results regarding mobile sensor vehicle control, obstacle detection and so on.

Matsummoto et al. [13] used multiple camera in order to control mobile robots. In their research, cameras are installed on their working space instead of mobile vehicle itself. Their whole system is consisted of mobile robots and multiple cameras and this helps the search of proper path of robots.

This work was supported by 2006 Hongik University Research Fund.

Their initial experiments were done with 3 pan-tilt cameras. Keyes et al. [14] researched various camera options such as lens type, camera type, camera locations and so on. They also used multiple cameras to obtain more precise information. In this paper we will provide MSV with multiple cameras in order to accomplish short distance, high accurate localization.

### C. Autonomous Driving Robot and dead-reckoning

This sort of localization is usually due to military area. For example DARPA, USA invests on unmanned vehicle, and their aim is about 30% of army vehicle without human on board controller. Stanley by Stanford university [15], which earned first prize in competitions, are equipped with GPS, 6 DOF gyroscope and can calculate the speed of driving wheels. Those sensors information can be combined to locate the position of their unmanned vehicle. They used computer vision system with stereo camera and single camera, and laser distance meter, radar in order to get environmental information. Sandstorm from CMU [16] is equipped with laser distance meter as a major sensor. Topographical model can be obtained by laser lines and the speed of car can be calculated by the density of laser line. Gimbal on their vehicle can install long distance laser scanner with seven laser sensors. Shoulder-mounted sensors can calculate height information of topography. Two scanners on bumpers can obtain obstacle information. Long distance obstacles can be identified by radar.

Our MSV are equipped with RSSI devices, stereo cameras and other sensors for dead-reckoning. Apart from the examples of locomotive robots, these equipments are for accurate localization.

## III. Mobile Sensor Vehicle

We developed MSV in order to experiment our localization method in real environment. Various versions of MSV are designed and implemented. The localization functions implemented on MSV are as follows:

- Long distance low accuracy localization by RSSI
- Medium distance medium accuracy localization by dead-reckoning tracking
- Short distance high accuracy localization by Stereo camera with computer vision.

In the following subsection we will discuss hardware and software of MSV respectively.

### A. Hardware

MSV is actually a mobile sensor node for MSN. Each MSV can move autonomously and can identify obstacles. They can communicate each other by 802.11 networking devices. The chassis of MSV are composed of aluminum composite with high durability and lightweight. The main driving mechanism is caterpillar composed of three wheels, L-type rubber belt, gears as shown in figure 1. The adoption of caterpillar is for minimization of driving errors. There are a lot of rooms to install additional sensor hardware. With digital compass equipped on the top of MSV, the accurate vehicle location can be sensored. This angular information can help exact localization of MSVs. The design concepts of MSV are as follows.

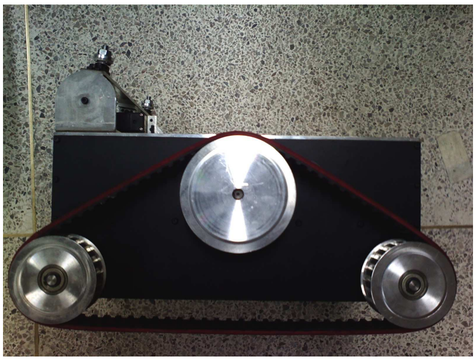


Fig. 1. Driving Mechanism Outlook

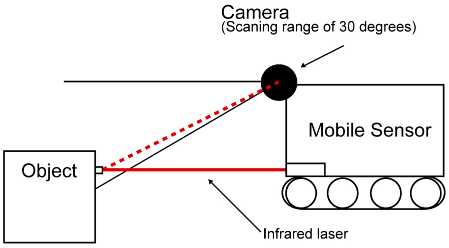


Fig. 2. Obstacle Detection Mechanism

- Autonomous mobility
- Extensibility of equipped sensors
- precise movement and mobility trail

And MSV characteristics as a node of mobile sensor network are as follows:

- Self identification and colleague identification
- Wireless communication
- Computing power

For autonomous driving, MSV must identify obstacles and avoid them. We use an infrared laser and cameras with infrared filter. IR laser is constantly lighting in parallel to round. Camera looks down grounds in a degree of 30 which is determined by experiments. The concept of this obstacle detections is depicted in figure 2. Obstacle reflects IR laser and sensed by camera [3], [5]. The obstacles with reflected IR will be detected as white lines. For short distance obstacles within the dead angle of camera, ultrasonic sensors are located under the MSV and in front of MSV. For computer vision based localization, MSVs are equipped with stereo eyes as shown in figure 3. Three servo motors can control two cameras independently. This stereo camera system can be used not only for localization but also for obstacle detection with diminished dead angle. There are three infrared LEDs mounted in the front of MSV. These LEDs are for computer vision based localization.

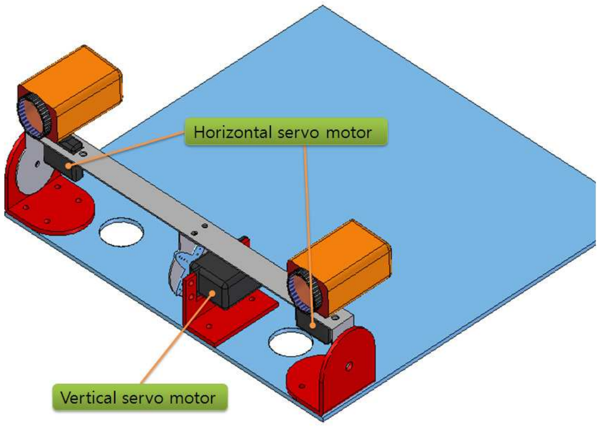


Fig. 3. Stereo Eye System

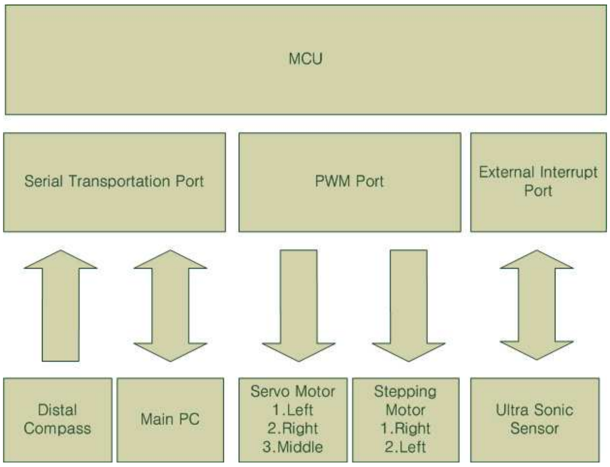

MCU
Serial Transportation Port
PWM Port
External Interrupt Port
Distal Compass
Main PC
Servo Motor 1.Left 2.Right 3.Middle
Stepping Motor 1.Right 2.Left
Ultra Sonic Sensor

Fig. 4. Hardware Structure

For hardware construction, we need micro controller unit, a serial communication port, a PWM port, an interrupt port in order to control motors and communicate with sensors. Figure 4 shows the conceptual structure of MSV hardware.

### B. Software

Software for MSV operations are required in a form of embedded software. Figure 5 shows required facility and their structure for MSV software.

Total part of software can be divided into five categories. One of the role of software is to convert sensor information into driving information. Information for driving can be obtained via serial communication from T-board (MCU) with driving information and angular information.

The location of MSV is constantly updated with the moving distance and updated angle. Camera class provides obstacle information as well as basic information for map building. Map building class builds a map with the information from T-board class and camera class. These maps are required for autonomous locomotion and localization. Network class provides networking functionalities between MSVs.

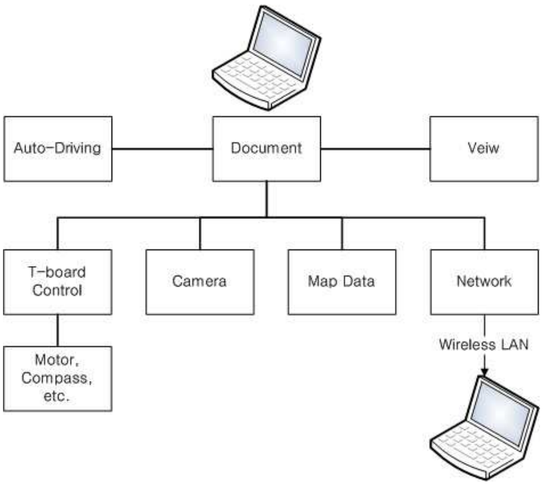


Fig. 5. Software Structure

*1) Core Software:* We implement core software based on multi threads. There is document class, which provides organic data flow between classes. Thus the major role of MSV software is as follows.

- Autonomous driving
- Motor control and driving distance identification
- Communication with MCU (T-board)
- Obstacle detection
- Map building
- Internetworking
- User interface dialog (Monitoring Program)

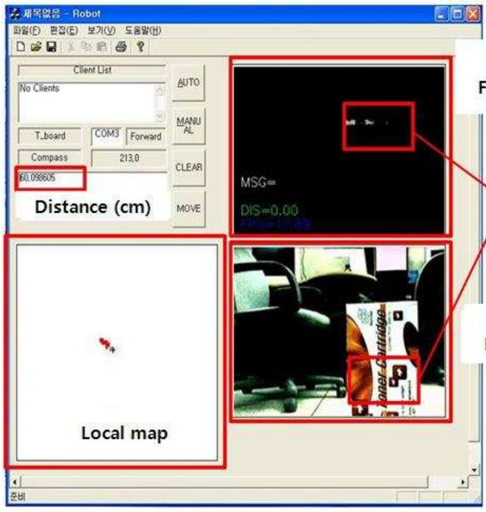
*2) Monitoring Program:* Monitoring program is a user interface between MSV and user. Monitoring program shows MSV condition, camera view, driving information, map built, and other sensor information. It also provides manual operation functionality of MSV. Figure 6 shows outlook of monitoring program. A mouse click on local map can drive MSV into dedicated location on the map.

Subdialog box in figure 7 shows the map built during the navigation of each MSV. It shows a result of localization based on dead-reckoning.

*3) Map Building:* Map building is one of the core parts of localization. The result of localization must be presented on local map and therefore be transferred to global map. MSVs communicate with each other in order to combine local maps into global maps. The following information will be shown on a map.

- White : Untapped territory
- Red : Territory with obstacle
- Blue : Territory with MSV
- Green : Tapped territory
- Undefined : Totally unknown territory

For map building we must consider relative coordinate and absolute coordinate. For example, obstacle information identified by MSV is in a form of relative coordinate. In relative coordinates, the very front of MSV is in angle 0 as shown in figure 8. This coordinate must be transformed into

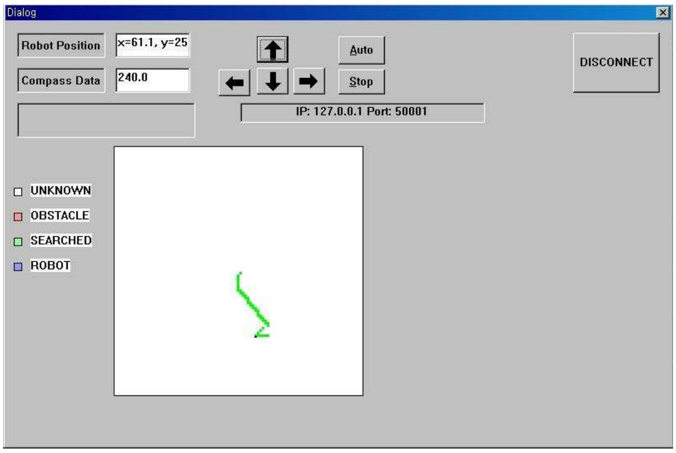


T_board COM3 Forward

Compass 213.0

60.098605

Fig. 6. Monitoring Program Screen



Robot Position x=61.1, y=25

Compass Data 240.0

IP: 127.0.0.1 Port: 50001

Fig. 7. Subdialog Program

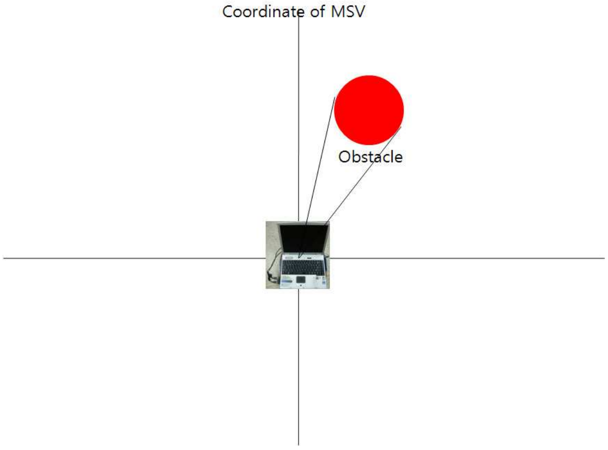


Fig. 8. Relative Coordinate

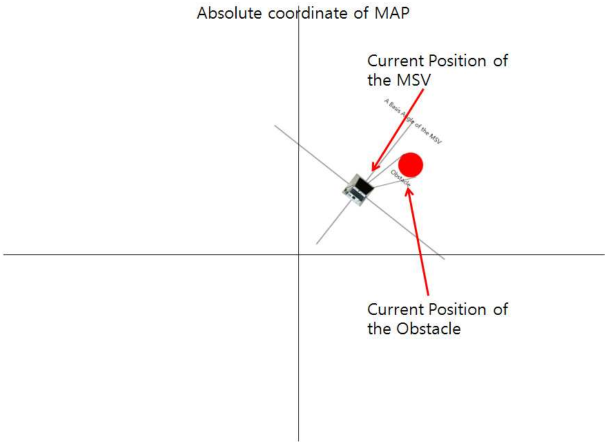

Absolute coordinate of MAP
Current Position of the MSV
Current Position of the Obstacle

Fig. 9. Absolute Coordinate

absolute coordinate as shown in figure 9 and therefore can be a part of map.

Local map is usually in a form of grid map. However in case of global map with huge capacities, grid map is very inefficient. Therefore we will use topological map for global map as presented by Kuipers and Bynn [4]. Thrun [6] presented a hybrid approach of both maps and we will consider it as our ultimate format of global map. Table I compares advantages and disadvantages of grid and topological map.

## IV. Sensor Mobility Trail based localization

### A. Dead-Reckoning

For the medium distance localization, we decided to utilize mobility trail. We define the range of medium distance between 4 meters and 40 meters since our vision based short distance localization covers within the range of 5 meters and RSSI based long distance localization is effective outside the range of 30 meters. Our aim is to trail the mobility of MSV and to record the trail on the local map with reasonable accuracy for medium distance localization. Every driving mechanism for mobile sensors or even mobile robots has mechanical errors and it is impossible to avoid such errors practically. We can summarize the cause of driving errors as followings:

- The difference between the sizes of two (left and right) wheel
- The distortion of wheel radius, i.e. the distance between average radius and nominal radius
- The wheel misalignment
- The uncertainty about the effective wheelbase
- The restricted resolution of driving motors (usually step motors)

Usually those errors are cumulated and final result will be void without proper error correction technique. However, in order to cope with those location errors due to mechanical errors, a method of dead-reckoning have been widely used and we also adopt such technique as well. Dead-reckoning is a methodology that calculates the moving distance of two wheels of MSV and derive the relative location from the origin of MSV.

Among the various versions of Dead-reckoning techniques, we used UMBmark technique from University of Michi-

TABLE I
COMPARISON BETWEEN GRID MAP AND TOPOLOGICAL MAP

| MAPS | Grid MAP | Topological MAP |
|---|---|---|
| Advantages | • precise presentation of geography of environment<br>• ease of algorithm design : environmental modeling, path finding, localization by map-matching | • simple presentation of environment and simple path planning<br>• tolerance of low accuracy mobile sensors<br>• natural interface to users |
| Disadvantages | • difficulty in path planning<br>• requirement of large memory and computation<br>• poor interface to symbolic problem solver | • impossibility of large map building with inaccurate, partial information<br>• difficulties in map-matching : difficulties in calculation of pivot sensor value<br>• difficulties in dealing complex environment |

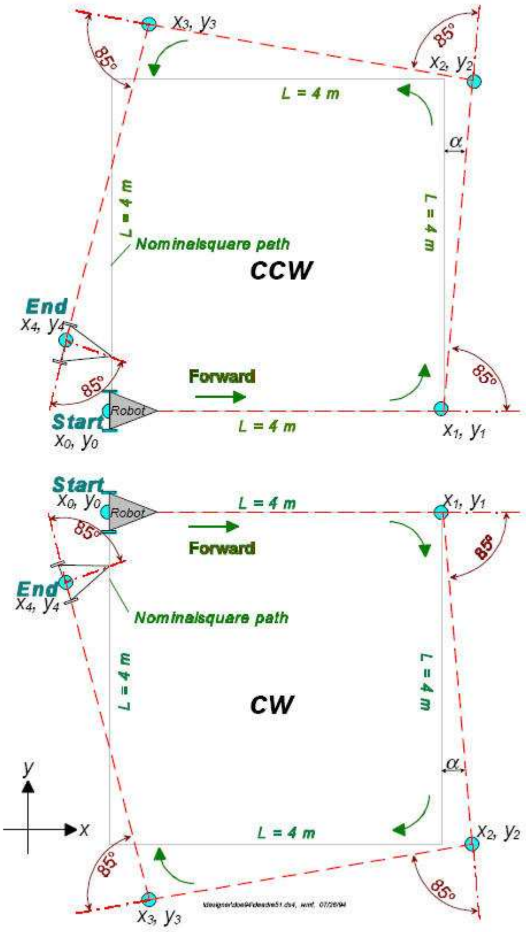


Fig. 10. Rotational Angle Error

gan [2]. UMBmark analyzes driving mechanism errors and minimized the effect of driving errors. UMBmark analyzes the result of MSV driving in a certain distance and compensates mechanical errors of MSV driving mechanism. The driving results of rectangular course, both in clockwise(CW) and counter-clockwise(CCW), and then analyzed.

Two error characteristics are classified in Rotation angle error and Wheel mismatch error. Rotational angle errors are for the difference between actual wheel sizes and theoretical design sizes of wheels. Due to rotational angle errors, CCW driving after CW driving shows larger errors as usual. For example, actual wheel size bigger than designed wheel size results in insufficient rotation at corners and then rotational angle errors are cumulated for the whole driving. The following equation summarizes the rotational angle error which is depicted in figure 10.

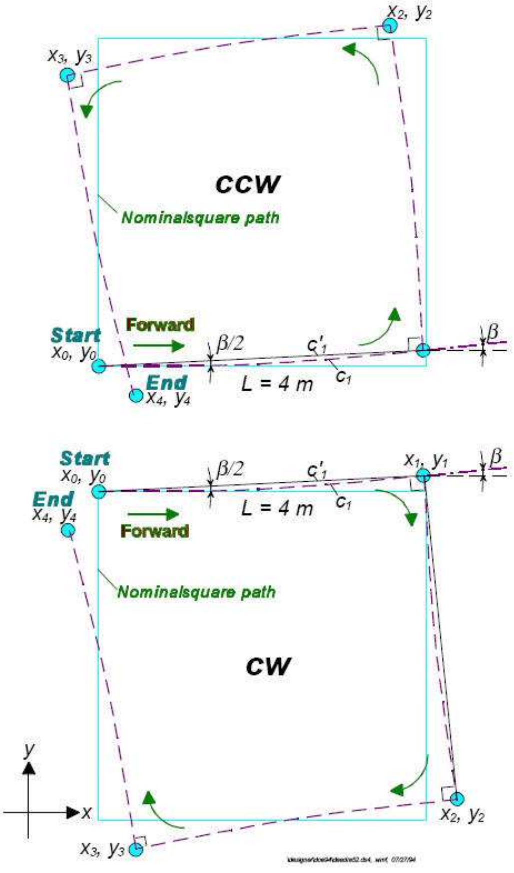


Fig. 11. Wheel Mismatch Error

$$E_d = \frac{D_R}{D_L}$$

where $D_R$ is diameter of left wheel and $D_R$ is diameter of right wheel. In short, $E_d$ is a ration between diameters of left wheel and right wheel.

Wheel mismatch errors are from wheelbase mismatch. This error causes skews in straight driving. With wheel mismatch error, the error characteristics of CW driving is opposite to CCW driving. The following equation summarizes the wheel size error which is depicted in figure 11.

$$E_b = \frac{90°}{90° - \alpha}$$

where $\alpha$ is a value of rotational angle error. $E_b$ stands for a ration between ideal and practical errors in rotation, i.e. wheel base error.

Equations above and figures 10 and 11 have been reprinted from [2].

Mechanical errors are systematical errors and therefore can be predicted and analyzed, while non-mechanical errors cannot be predicted because non-mechanical errors are due to the driving environment. Non-mechanical errors are classified as follows:

- Uneven driving floor or ground
- Unpredicted obstacle on driving course
- Slipping while driving

We applied UMBmark to our MSV and the following subsection shows the result.

### B. Driving Error Correction of MSV

We composed a set of experiment for MSV driving in order to apply UMBmark. The driving experiments have been made on the flat and usual floor with the rectangular driving course of 4 × 4 meters. As shown in figures 10 and 11, both CW and CCW driving have been made and error values have been measured. These error values are incorporated in our software system and MPU controllers.

With the following equations from [2] we can find the error value for error correction.

$$b_{actual} = E_b \times b_{nominal}$$

where $b_{actual}$ is an actual wheelbase and $b_{nominal}$ is a measured wheelbase.

$$\Delta U_{L,R} = c_{L,R} \times c_m \times N_{L,R}$$

Where U is actual driving distance, N is the number of pulses of the encoder, and $c_m$ is the coefficient to convert pulse per centimeters.

Our experimental result with driving location correction by UMBmark dead-reckoning mechanism is shown in figure 12. Circled dotes are result from CCW driving and rectangular dotes are from CW driving. Empty dotes are of uncorrected driving results while filled dotes are of driving results with UMBmark in 16 meter driving experiment. Note that the origin of MSV (starting point) at the coordinate (0,0) are at the upper right part of the figure. Without dead-reckoning technology, MSV returns to erroneous point than the origin point, at the left part of the figure. This MSV tends to show more errors with CW driving. With the application of UMBmark technique, we achieved faithful result within 10 centimeters of error range in total. Directional errors are within the range of 3 centimeters from the origin. Since our approach is for mechanical driving errors, non-mechanical errors can be avoided and thus we will introduce real-time correction of driving with the help of digital compass for the future researches. Thus it is possible to mention that the trail of MSV in figure 7 is in the correct location within the errors of 10cm in our experimental environments.

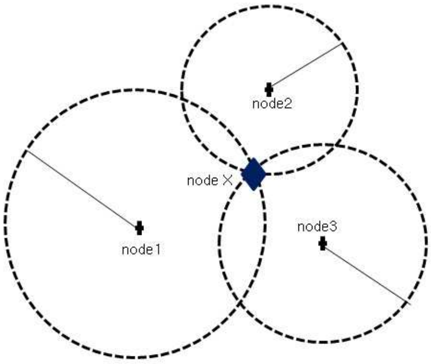


Fig. 13. Triangulation With RSSI Measurement

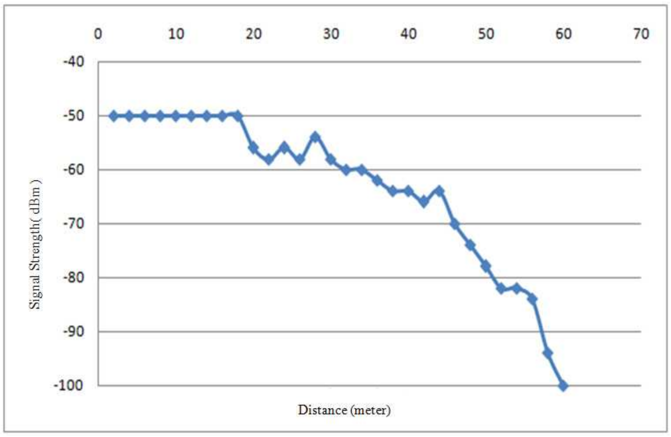


Fig. 14. RSSI values According To Distance

## V. RSSI BASED LOCALIZATION

Our MSV are equipped with homogeneous 802.11 networking devices with RSSI facilities. With distance information we can do triangulation with at least three nodes and one anchor. Our monitoring station with monitoring program can act as an anchor. The 802.11 networking devices can be switched to AP (Access Point) mode so that each MSN can act as AP. With software modification that utilizes 802.11 device RSSI features, we can achieve RSSI based localization for our MSN.

The unit of RSSI is in dBm (-50dBm $\sim$ -100dBm) and it designates distances between specified MSVs. As already mentioned, the RSSI value is very affective by environments, and we obtain error rate of 10 $\sim$ 15% in specific distance. The RSSI value is very sensitive with hardware vendor and the direction of AP [7]. Our experimental environment is as follows.

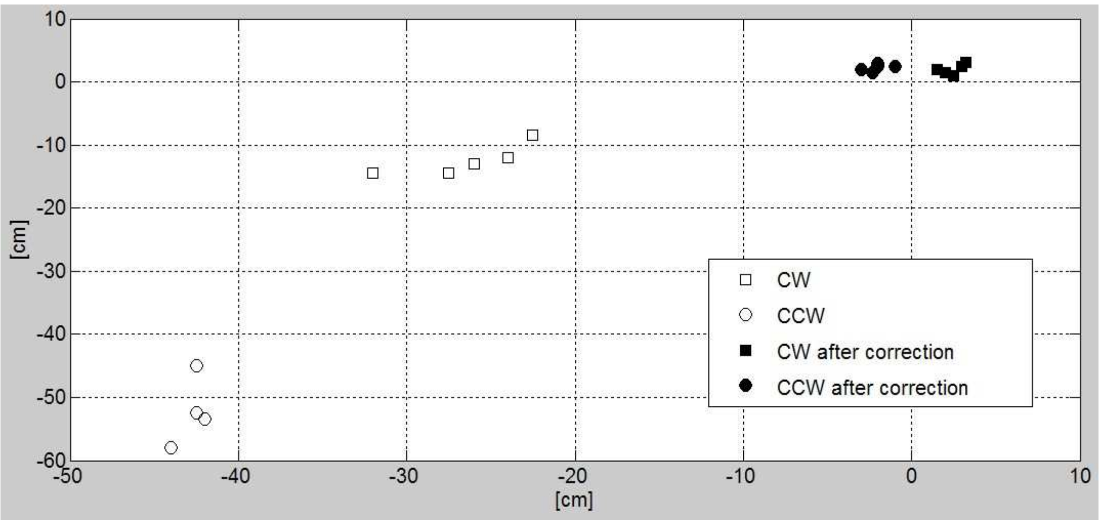


Fig. 12. Result of UMBmark

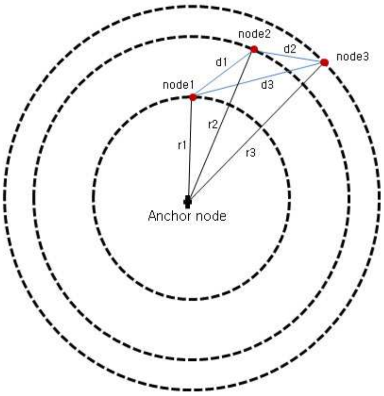


Fig. 15. Relation Between One Fixed Anchor And Mobile Sensor Vehicles

- Wide open area
- Intel Wireless LAN 2100 3B Mini PC Adapter
- WRAPI software model [11]
- One fixed anchor as monitoring station

For the calibration of our RSSI device, a set of experiment has been conducted and the results are shown in figure 14.

We can find within the distance of 20 meter, RSSI is no more useful for distance measure since the signal strength is too high. Our experiments shows RSSI based localization is useful more than 35m distance. The values are within error range of 15% by experiment. This is the main reason why we choose RSSI based localization for long distance, low accuracy localization.

From the distance information from RSSI sensing, we can do triangulation as shown in figure 13. For actual implementation, we have one fixed anchor and can do more precise localization with a known anchor coordinate as shown in figure 15. The figures show three mobile nodes one anchor node. The distance obtained from circle $r1, r2, r3$ can be obtained from RSSI values. Thus with this environment we can triangulate the coordinate node $X$ from the intersection of circles drawn by node 1, node 2, and node 3 [9][10].

Thus from the distance which can be obtained from RSSI values, let the distance be d$i$ from radius of circle r$i$ The following algorithm 1 shows a procedure to find coordinates of each MSV with provided distance information by RSSI.

Figure 16 shows a final result in RSSI based localization. The x-axis stands for actual distance between MSVs and y-axis shows a distance calculated by algorithm 1. As we predicted the RSSI based localization is useful with the distance more than 30 meters. On the range where RSSI based localization is effective, we can see errors between actual distance and calculated distance. We believe it is tolerable since we have another method of localization with more accuracy within the distance of 30 meters. Of course, the distance information is not a sufficient condition for localization. The other information of direction of MSV can be obtained by digital compass on each MSV. Thus we implemented long distance, low accuracy localization.

## VI. Conclusions

For the localization methodologies for mobile sensor network, we proposed three different categories of localization methodology. In addition for the experiment, we implemented mobile sensor vehicle as a node of mobile sensor network. We showed brief description of mobile sensor vehicle including hardware and software functionalities. The driving mechanism hardware and software cooperate with each other and naturally achieve localization based on dead-reckoning, which is a medium distance and medium accuracy localization. The result of localization can be presented on local maps and eventually be merger into global maps. In addition we showed RSSI based localization. The long distance, low accuracy localization can be implemented by 802.11 networking devices

**Algorithm 1** Localization of Sensor Nodes with RSSI Measurement

```
Input : d1, d2, d3, r1, r2, r3
//Distance d1, d2, d3
//Circle r1, r2, r3
Output :  SolutionList
LinkedList SolutionList
//Mobile Sensor Node Coordinates

for (each (x1,y1) on Circle r1)
{
 for (each (x2,y2) on Circle r2)
 {
  if (d1 ==
   distance between (x1,y1) and (x2,y2))
  {
   for(each (x3,y3) on Circle r3)
   {
    if (d2 ==
     distance between (x2,y2) and (x3,x3))
    {
     if (d3 ==
     distance between (x3,y3) and (x1,y1))
     {
      SolutionList =
       Coordinate (x1,y1),(x2,y2),(x3,y3)
}}}}}}
```

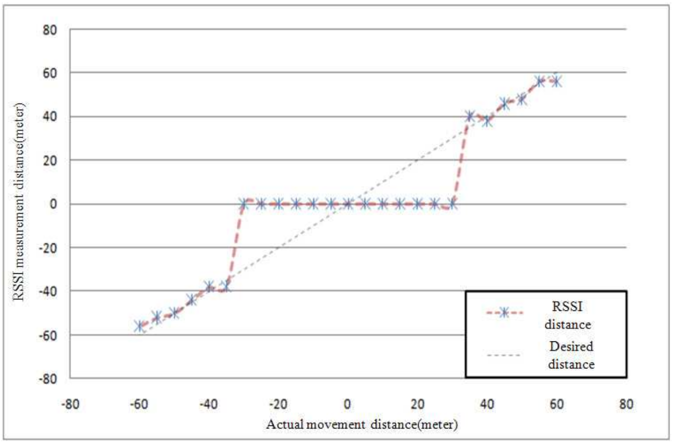


Fig. 16. Actual and RSSI based Distance

and algorithms based on triangulation. For short range, high accuracy localization, a stereo camera system is utilized. The detailed result of this computer vision based approach can be found in [18]. From these three levels of localization, we believe that we implemented useful localization system and will do more research using this platform. For example multiple MSV can cooperate and communicate each other and then a formation based on localization can be made. Also, a digital compass is also equipped in each MSV. Our medium accuracy localization can be more precise with more error correction mechanism with digital compasses.

## References

[1] Localization for Mobile Sensor Networks, Lingxuan Hu and David Evans, Tenth Annual International Conference on Mobile Computing and Networking (MobiCom 2004). Philadelphia, 26 September – 1 October 2004.

[2] J. Borenstein, L. Feng, D. K. Wehe, Y. Koren, Z. Fan, B. Holt, B. Costanza, UMBmark—A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots 1995.

[3] R. B. Fisher, A. P. Ashbrook, C. Robertson and N. Werghi, A low-costrange finder using a visually located, structured light source. "3-D Digital Imaging and Modeling", 1999. *Proceedings. Second International Conference on 4-8 Oct. 1999* Pages: 24–33.

[4] B. Kuipers and Y.-T.Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," Journal of Robotics and Autonomous Systems, 1991.

[5] J. Ollero Gonzalez and A. Reina, "Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1904–1909, 1994.

[6] Zhengyou Zhang , "A Flexible New Technique for Camera Calibration." *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 22, No. 11, November 2000.

[7] Martin A. Fischler and Robert C. Bolles, SRI International. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." Communications of the ACM, June 1981.

[8] J. Borenstein and Y. Koren, "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance, IEEE TRAN." *Robotics and Auto*, Vol. 7, No. 4, 1991.

[9] Anastasia Katranidou, "Location-sensing using the IEEE 802.11 Infrastructure and the Peer-to-peer Paradigm for Mobile Computing Applications," Heraklion, February 2006.

[10] P. V. Bahl and Padmanabhan, "Radar: An In-Building RF-based User Location and Tracking System," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), March 2000.

[11] WRAPI, http://sysnet.ucsd.edu/pawn/wrapi

[12] IEEE Std. 802-11.1997, "IEEE Standard for Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specification," June 1997.

[13] Kohsei Mstsumoto, Jun Ota and Tamio Arai, "Multiple Camera Image Interface for Operating Mobile Robot," Proceedings of the 2003 IEEE lnternational Workshop on Robot and Human Interactive Communication Millbrae, California, USA, Oct. 31 – Nov. 2, 2003.

[14] Brenden Keyes, Robert Casey, Holly A. Yanco, Bruce A. Maxwell, Yavor Georgiev, "Camera Placement and Multi-Camera Fusion for Remote Robot Operation," IEEE Int'l Workshop on Safety, Security and Rescue Robotics, Gaithersburg, MD, August, 2006.

[15] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, Pamela Mahoney, "Stanley: The Robot that Won the DARPA Grand Challenge." Journal of Field Robotics 23(9), 661–692 (2006).

[16] A. Elfes, "Occupancy grids: A Probabilistic Framework for Robot Perception and Navigation," PhD thesis, Department of Electrical and computer Engineering, Carnegie Mellon University, 1989.

[17] Magnus Lindhe, Karl Henrik Johansson, "Communication-aware trajectory tracking," IEEE International Conference on Robotics and Automation, 2008, , pp. 1519-1524, 2008, IEEE.

[18] Kwansik Cho, Ha Yoon Song, Jun Park, "Accurate Localization in Short Distance based on Computer Vision for Mobile Sensors," First International Symposium on Multimedia—Applications and Processing (MMAP'08), Oct. 2008, IEEE.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 719–725

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Component-based Approach to Verification of Embedded Control Systems using TLA$^+$

Ondrej Rysavy and Jaroslav Rab

Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
Email: {rysavy,rabj}@fit.vutbr.cz

***Abstract*—The method for writing TLA$^+$specifications that obey formal model called Masaccio is presented in this paper. The specifications consist of components, which are built from atomic components by parallel and serial compositions. Using a simple example, it is illustrated how to write specifications of atomic components and components those are products of parallel or serial compositions. The specifications have standard form of TLA$^+$specifications hence they are amenable to automatic verification using the TLA$^+$model-checker.**

## I. Introduction

SOFTWARE running in embedded systems necessary acquires some properties of the physical world. Usually, these properties form a part of non-functional aspects in system requirements [1]. To model embedded software, these aspects must be considered by a specification method otherwise the model of a system easily diverges from the reality and becomes inapplicable in further refinement and analysis. Constructing large systems relies on effective and systematic application of modular approach. A large class of entities playing the role of building blocks that can be composed have been defined, most notably, classes for object-oriented program construction, components in hardware design, procedures and modules in procedural programming, and active objects and actors for reactive programming [2].

This paper deals with a method based on a formalism called Temporal Logic of Actions [3] that enables to describe embedded control software in a modular manner and apply an automatized model-checker tool to verify required properties of a specification. The main contribution of this paper lies in demonstration of how the TLA$^+$specifications whose interpretation is that of a formal model called Masaccio[4] can be written in a systematic way. The formal model permits to construct a hierarchical definition of components that are built from atomic components using operations of parallel composition, serial composition, renaming of variables, renaming of locations, hiding of variables, and hiding of locations. As the resulting TLA$^+$specifications have the form of a conjunction of initial predicate and next-state actions, they are readily explorable by the TLA$^+$explicit model-checker.

## II. Component Model

This section gives a brief overview of a formal model for embedded components as defined by Henzinger in [4]. In this paper, only discrete components are considered, although the proposed approach relies on the language that can be applied to hybrid systems [5] as well.

A fundamental entity of the model is a component. The component structures the system into architectural units that interact through defined interfaces. It is possible to structure components into a hierarchy that can be arbitrary nested to simplify the system design. The component $A$ consists of definition of interface and internal behavior. An interface defines disjoint sets of input variables, $V_A^{in}$, output variables, $V_A^{out}$, and a set of public locations, $L_A^{intf}$. An execution of the component consists of a finite sequence of jumps. A jump is a pair $(p, q) \in [V_A^{in,out}] \times [V_A^{in,out}]^1$. An observation $p$ is called the source of jump $(p, q)$ and an observation $q$ is called the sink of jump $(p, q)$. A jump $v$ is successive to jump $u$ if the source of jump $v$ is equal to the sink of jump $u$. Formally, an execution of $A$ is a pair $(a, w)$ or a triple $(a, w, b)$, where $a, b \in L_A^{intf}$ are interface locations and $w = w_0 \dots w_n$ is a nonempty, finite sequence of jumps of $A$ such that every jump $w_i$, for $1 \leq i \leq n$ is successive to the immediately preceding step $w_{i-1}$. We write $E_A$ for the set of executions of the component $A$.

*An atomic component* is the simplest form of components found in Masaccio. The behavior of the component is solely specified by its jump action. The interface of atomic component exploits input variables read by the component and output variables controlled by the component. The component $A(J)$ has two interface locations, $from$ and $to$; that is, $L_{A(J)}^{intf} = \{from, to\}$. The entry condition of $from$ is the projection of the jump predicate to the unprimed I/O variables. The entry condition of *to* is unsatisfiable.

Two components $A$ and $B$ can be combined to form a *parallel composition* $C = A \otimes B$ if the output variables of $A$ and $B$ are disjoint and for each interface location $a$ common to both $A$ and $B$, the entry conditions of $a$ are equivalent in $A$ and in $B$. The input variables of component

[1] The research has been supported by the Czech Ministry of Education in the frame of Research Intentions MSM 0021630528, and by the Grant Agency of the Czech Republic through the grants GACR 201/07/P544 and GACR 102/08/1429.

[1] $[V_A^{in,out}]$ stands for a set of all possible assigments of values into input and output variables of a component.

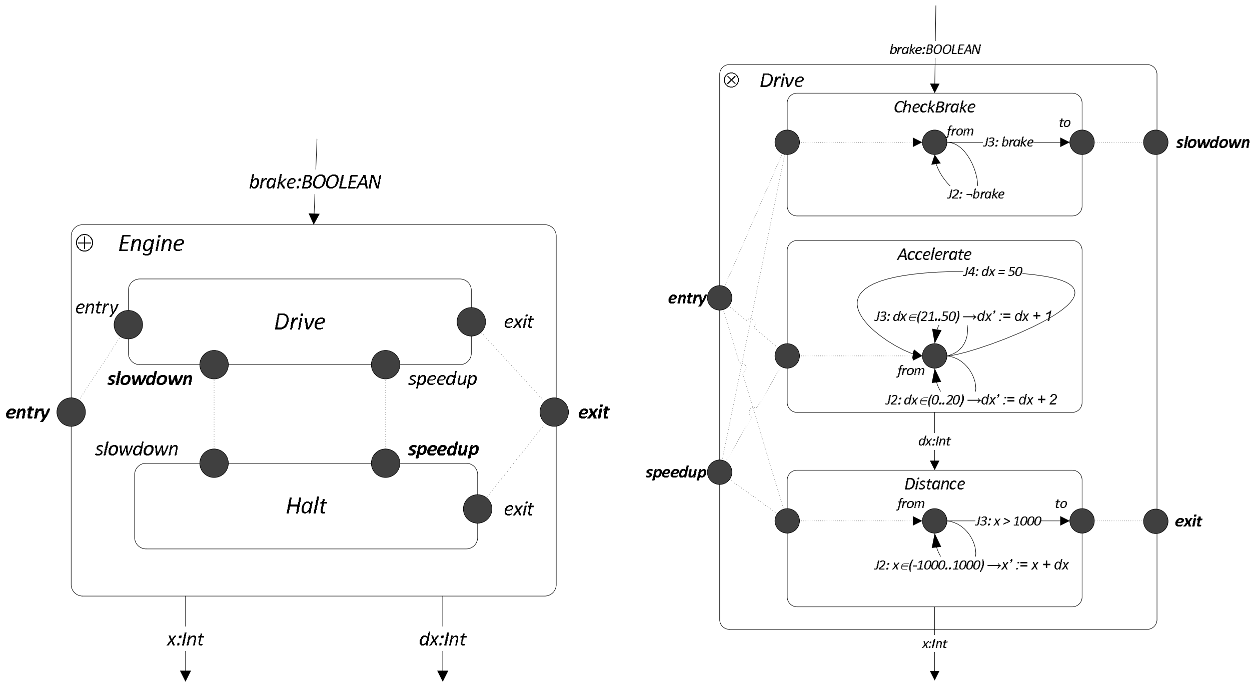


Fig. 1. The components Engine and Drive

$V_C^{in} = (V_A^{in} \setminus V_B^{out}) \cup (V_B^{in} \setminus V_A^{out})$. The output variables of the component are $V_C^{out} = V_A^{out} \cup V_B^{out}$. The interface locations of $A \otimes B$ are the interface locations of $A$ together with the interface locations of $B$. An interface location $a$ that is common to $A$ and $B$ and its entry conditions agree in both components has this entry condition also in $A \otimes B$. Other interface locations cannot be used to entry the component.

The definition of parallel composition specifies that each jump is done in synchronous manner in both subcomponents. Moreover, if one component reaches the exit interface location then the execution in the other component must be terminated. If both components reach their exit locations one is chosen nondeterministically. As the consequence of these properties, parallel composition operation is associative and commutative.

Two components $A$ and $B$ can be composed in series to form a *serial composition* $C = A \oplus B$ if the set of output variables are identical; that is, $V_A^{out} = V_B^{out}$. The input variables of composed component is $V^{in} = V_A^{in} \cup V_B^{in}$. The interface locations of $A \oplus B$ are the interface locations of $A$ together with the interface locations of $B$. If $a$ is an interface location of both $A$ and $B$, then the entry condition of $a$ in $A \oplus B$ is the disjunction of the entry conditions of $a$ in the subcomponents $A$ and $B$.

The set of execution of the component $C = A \oplus B$ contains 1) the pair $(a, w)$ iff either $(a, w|_A)$ is an execution of $A$, or $(a, w|_B)$ is an execution of $B$, 2) the triple $(a, w, b)$ iff either $(a, w|_A, b)$ is an execution of $A$, or $(a, w|_A, b)$ is an execution of $B$. The operator of serial composition is associative, commutative, and idempotent.

To support these two compositional operations, the renaming and hiding operations are defined for variables and locations. The renaming operation maps variables and locations of different names to each other that allows for sharing data and control between components. Hiding makes variables or locations internal to the component, which is useful when a complex behavior is modeled inside the component.

## III. TLA$^+$

Temporal Logic of Actions (TLA) is a variant of linear-time temporal logic. It was developed by Lamport [3] primarily for specifying distributed algorithms, but several works shown that the area of application is much broader. The system of TLA+ extends TLA with data structures allowing for easier description of complex specification patterns. TLA+ specifications are organized into modules. Modules can contain declarations, definitions, and assertions by means of logical formulas. The declarations consist of constants and variables. Constants can be uninterpreted until an automated verification procedure is used to verify the properties of the specification. Variables keep the state of the system, they can change in the system and the specification is expressed in terms of transition formulas that assert the values of the variables as observed in different states of the system that are related by the system transitions. The overall specification is given by the temporal formula defined as a conjunction of the form $I \wedge \Box[N]_v \wedge L$, where I is the initial condition, N is the next-state relation (composed from transition formulas), and L is a conjunction of fairness properties, each concerning a disjunct of the next-state relation. Transition formulas, also called actions, are ordinary formulas of untyped first-order logic defined on a denumerable set of variables, partitioned into sets of flexible and rigid variables. Moreover, a set of primed flexible variables, in the

```
┌──────────────────── MODULE Accelerate ────────────────────┐
EXTENDS Naturals
VARIABLES dx, clock, location
├───────────────────────────────────────────────────────────┤
J1 ≜ ∧ location = "from" ∧ location′ = "from"
     ∧ dx ∈ (0 .. 20) ∧ dx′ = dx + 2
     ∧ clock′ = ¬clock

J2 ≜ ∧ location = "from" ∧ location′ = "from"
     ∧ dx ∈ (21 .. 49) ∧ dx′ = dx + 1
     ∧ clock′ = ¬clock

J3 ≜ ∧ location = "from" ∧ location′ = "from"
     ∧ dx = 50 ∧ dx′ = 50
     ∧ clock′ = ¬clock

Init  ≜  ∧ dx ∈ (0 .. 50) ∧ clock ∈ BOOLEAN  ∧ location = "from"

Next ≜ J1 ∨ J2 ∨ J3
└───────────────────────────────────────────────────────────┘
```

Fig. 2. The TLA$^+$ specification of component Accelerate

form of $v'$, is defined. Transition formulas then can contain all these kinds of variables to express a relation between two consecutive states. The generation of a transition system for the purpose of model checking verification or for the simulation is governed by the enabled transition formulas. The formula $\Box[N]_v$ admits system transitions that leave a set of variables v unchanged. This is known as stuttering, which is a key concept of TLA that enables the refinement and compositional specifications. The initial condition and next-state relation specify the possible behaviour of the system. Fairness conditions strengthen the specification by asserting that given actions must occur. The TLA+ does not formally distinguish between a system specification and a property. Both are expressed as formulas of temporal logic and connected by implication $S \implies F$, where S is a specification and F is a property. Confirming the validity of this implication stands for showing that the specification S has the property F. The TLA+ is accompanied with a set of tools. One of such tool, the TLA+ model checker, TLC, is state-of-the-art model analyzer that can compute and explore the state space of finite instances of TLA+ models. The input to TLC consists of specification file describing the model and configuration file, which defines the finite-state instance of the model to be analysed. An execution of TLC produces a result that gives answer to the model correctness. In case of finding a problem, this is reported with a state-sequence demonstrating the trace in the model that leads to the problematic state. Inevitably, the TLC suffers the problem of state space explosion that is, nevertheless, partially addressed by a technique known as symmetry reduction allowing for verification of moderate size system specifications.

## IV. Specification of Components

Using a simple example as required by space constraints, this section explains the construction of TLA$^+$specifications that corresponds to Masaccio embedded components.

An example represents a specification of component $Engine$ taken from [4]. This component is a part of a complex specification that models the control of a railway crossing. In particular, the $Engine$ component controls acceleration and deceleration of a train that is moving in a near distance to the railway crossing. Although this example is rather trivial, it is sufficient to demonstrate basic principles of the specification method as it contains both parallel and serial compositions.

The component $Engine$ and its subcomponents are visually modeled in figure 1. Components are represented by rectangles. Input and output variables are represented by arrows connected to component boundaries. Locations are represented by solid discs. Jump actions are represented by arrows. A jump is labeled with condition predicate and action predicate, which computes new values of output variables.

Component $Engine$ consists of a serial composition of two subcompoments, namely, $Drive$ and $Halt$. An entry location is directly connected with one of the locations of $Drive$ component. There is one exit location that is accessible from both subcomponents. Other interface locations, namely $slowdown$ and $speedup$, serve for passing the control flow between $Drive$ and $Halt$ components. The component interacts with the environment by reading input variable $brake$ and controlling output variables $x$ and $dx$. These variables are also available to both subcomponents.

Component $Drive$ governs train acceleration. The component is a parallel composition of three atomic components: $CheckBrake$, $Accelerate$, and $Distance$. Input variable $brake$ determines whether the train accelerates or decelerates. Its value is observed by $CheckBrake$ component that takes away control from $Drive$ component if variable $brake$ signalizes the application of train's brake. In component $Accelerate$, the actual speed of the train is computed. The train dynamics is simplified by considering that the train accelerates by $1ms^{-2}$ if its velocity is greater $20ms^{-1}$ and by $2ms^{-2}$ if its velocity is

```
┌──────────────────────────── MODULE Drive ────────────────────────────┐
EXTENDS Integers, Sequences
VARIABLE brake, x, dx, clock, loc1, loc2, loc3
├──────────────────────────────────────────────────────────────────────┤
driveBrake ≜ INSTANCE DriveBrake WITH location ← loc1
accelerate  ≜ INSTANCE Accelerate WITH location  ← loc2
distance ≜ INSTANCE Distance WITH location ← loc3
├──────────────────────────────────────────────────────────────────────┤
Init  ≜ driveBrake!Init  ∧ accelereate!Init  ∧ distance!Init

Next ≜ driveBrake!Next ∧ accelereate!Next ∧ distance!Next
└──────────────────────────────────────────────────────────────────────┘
```

Fig. 3. The TLA$^+$ specification of component $Drive$

less than $20ms^{-1}$, respectively. Finally, component $Distance$ is responsible for computing the actual distance from the railway crossing.

Component $Halt$ has similar structure to component $Drive$. It's purpose is to slow the train down as long as input variable $brake$ is set to true. If $brake$ is released it passes the control back to $Drive$ component through location $speedup$.

To show that TLA$^+$ specifications conform to Masaccio interpretation, the interpretation of TLA$^+$ expressions needs to be defined. The following simplified system is used (for complete semantics see e.g. [6]). The TLA$^+$ module is called a standard module if it has the form of conjunction of an initial state predicate and a next-state action predicate. The meaning of a standard TLA$^+$ module $M = \langle V, I, N \rangle$, where $V$ is a finite set of variables, $I$ is an initial predicate, and $N$ is a set of next-state actions, is then defined by valuation function $\mathcal{V}_s(x)$, which assigns a value to each variable $x \in V$ and each state $s$, and model satisfying relation, $s \models_{\mathcal{M}} p$, which asserts that proposition $p$ is true in state $s$ in the model $\mathcal{M}$ of module $M$. A model $\mathcal{M}$ is a graph that consists of a set of nodes $\mathcal{M}_N$ representing states, and a set of edges $\mathcal{M}_E$ representing transitions between states. Obviously, a set of initial states of model $\mathcal{M}$ is defined as all states satisfying the initial predicate; that is, $\mathcal{I} = \{s \in \mathcal{M}_N : s \models_{\mathcal{M}} I\}$. Each next-state action $n$ can be split into a part $n_1$, where only unprimed variables occur, and a part $n_2$, where also primed variables occur. If $s \models_{\mathcal{M}} n_1$ and $r \models_{\mathcal{M}} n_2$ then, necessary, $s, r \in \mathcal{M}_N$ and $\langle s, r \rangle \in \mathcal{M}_E$. Masaccio interpretation is defined in terms of execution traces. Obviously, an execution is a trace that can be generated by traversing a graph $\mathcal{M}$. Formally, a trace $w$ consists of jumps $(p, q)$, such that $p, q \in \mathcal{M}_N$ and $\langle p, q \rangle \in \mathcal{M}_E$.

### A. Specifying atomic components

According to Masaccio semantics, an atomic discrete component $A(J)$ is completely specified by a jump predicate that defines a set of legal jumps $J$. Further, an atomic component has an arbitrary number of input and output variables. In each atomic component, there are only two interface locations, denoted as $from$ and $to$.

The representation of atomic component is straightforward in TLA$^+$ language. In figure 2, TLA$^+$ description of $Accelerate$ component is shown. A set of jumps is a conjunction of three next-state actions. Action $J1$ represents acceleration of a train in lower speeds. Action $J2$ represents acceleration of the train in higher speeds. Finally, action $J3$ specifies that if the train reaches its maximal speed it maintains this speed. In addition to state variable $location$ and controlled variable $dx$, which keeps the actual train's speed, the module declares a boolean variable $clock$, which models the passing of the time. Introducing the system clock is necessary for synchronization of the components. While this simple approach seems to be appropriate in this case, more flexible approach, e.g. [7], might be considered in more involved real-time specifications.

The following definition generalizes atomic component specification rules.

*Definition 1 (atomic component):* An atomic component $A(J)$ is a TLA$^+$ module $M = \langle V, I, N \rangle$ such that:

- it declares a variable for each I/O variable of the atomic component; that is, $\forall v : T \in V^{in,out}_{A(J)} : \exists v \in V$ such that $s \models_{\mathcal{M}} v \in T$ for all $s \in \mathcal{M}_N$.
- it declares a location variable; that is, $location \in V$, and $s \models_{\mathcal{M}} v \in \{from, to\}$ for all $s \in \mathcal{M}_N$.
- the meaning of next-state action $N$ agrees with the predicate $\varphi^{jump}_J$; that is, $(p, q) \models_{\mathcal{M}} N$ if each unprimed variable $x$ from $N$ is assigned the value $\mathcal{V}_p(x)$ and each primed variable $y$ from $N$ is assigned the value $\mathcal{V}_q(y)$.
- the meaning of initial predicate $I$ agrees with the predicate $\varphi^{en}_{A(J)}(from)$; that is $p \models_{\mathcal{M}} I$ for every trace of atomic component $A(J)$ with prefix $(from, (p, q))$ if each variable $x$ from $I$ is assigned the value $\mathcal{V}_p(x)$.

As it can be seen from the TLA$^+$specification in figure 2, the atomic specification contains variable denoted as $clock$. This variable serves to synchronization purposes. It enforces that parallel actions are executed at the same time. Therefore all jumps include the condition stating $clock' = \neg clock$. Except proper actions, there are also specific actions supporting serial compositions as described later in this section. These specific actions violate this condition requiring that the time is stopped; that is, $clock' = clock$.

A component can be entered at location $a$ if an entry condition $\varphi^{en}_A(a)$ is satisfied at $(p, q')$; that is, $(p, q') \models \varphi^{en}_A(a)$. A valid expression of entry condition is similar to next-state relation in TLA. It has form of conjunctions of expressions that can contain unprimed and primed variables. Contrary to TLA, the entry condition is enabled if both unprimed and primed parts are satisfied in $(p, q)$, while the TLA action is enabled if the unprimed part is satisfied in state $p$. This is important for guarantee of the dead-lock free property. As TLC automatically checks whether the given specification is dead-lock free, it is possible to relax the entry condition into its weaker form $p \models \varphi_A^{en}(a)$, which does not contain the primed variables.

### B. Specifying Composition of Components

The component $Drive$ shown in figure 1 is a result of parallel composition of three subcomponents. The corresponding TLA$^+$ specification is given in figure 3. The semantics of parallel composition corresponds to joint-action specification as described by Lamport in [3, p.147]. Its encoding in TLA is straightforward.

The $Drive$ module contains input and output variables $brake$, $x$, $dx$ and also variables $loc1$, $loc2$ and $loc3$ that keeps the state of subcomponents $DriveBrake$, $Accelerate$ and $Distance$, respectively. These location variables are bound to variable $location$ in each component during the component instantiation as declared on lines 5-7. Line 9 defines a collection of initial states of the subcomponents. The initial predicate $Init$ is a conjunction of initial predicates of all subcomponents. Next-state action predicate $Next$ is a conjuction of next-state predicates of subcomponents, which gives the intended execution interpretation of the component; that is, the jumps of subcomponents are executed in parallel and in synchronous manner.

*Definition 2 (parallel composition):* A component $C = A \otimes B$ composed in parallel from subcomponents $A$ and $B$ can be written as TLA$^+$module $M_C = \langle V_C, I_C, N_C \rangle$, where

- $V_C$ is a set of module's variables that includes all input and output variables of the subcomponents and location variables (an implicit renaming of location variables is considered to prevent the slash of their names in module $M_C$); that is $V_C = V_A \cup V_B$.
- $I_C$ is an initial predicate that is a conjunction of initial predicates of both submodules and a component specific constraints; that is $I_C = I_A \wedge I_B \wedge I$.
- $N_C$ is a next-state action predicate that is defined as a conjunction of next-state predicates of both submodules; that is $N_C = N_A \wedge N_B$.

A state space of a composed component is generated according the initial predicates and next-state actions of its subcomponents. The conjunction of next-state actions requires that there are simultaneous jumps in each of the subcomponent. Moreover if one of the subcomponent reaches its end location, which causes that such component has not enabled action, it is not possible to execute any jump in any of the contained components. This configuration is then recognized as the end location of the component.

The serial composition of components requires that only one contained component has control at a time. This needs to be reflected in a location configuration. Therefore a special location, denoted as empty string (""), has been added to represent a state of a component without a control. A component whose location configuration is "" cannot execute any of its jumps. To enable the passing of control between components, specific actions that modify only location variables are added into the specification. Their purpose is similar to that of connector elements that can be found in many architecture description languages, e.g. [8].

The example of a component composed in series is shown in figure 4. The module $Engine$ instantiates two subcomponents, namely $Drive$ and $Halt$. The initial predicate specifies that the component $Drive$ will have control when component $Engine$ is first executed. This means to define valid initial interface location for subcomponent $Drive$, in particular, to assign value $from$ to $dl1$, $dl2$ and $dl3$ variables, and to define that $Halt$ subcomponent is in the idle state that is expressed by assigning "" to variables $hl1$, $hl2$ and $hl3$. To define next-state action predicates we assume that specification of $Halt$ and $Drive$ were both extended with the following definition:

$$Idle \triangleq loc1 = "" \wedge loc2 = "" \wedge loc3 = ""$$

This definition asserts that component is in the idle state. Therefore, action $L1$ and action $L2$ define a behavior of the containing component as an execution of component $Drive$ and component $Halt$, respectively, assuming that a complementing component is being idle during this execution. Finally, two connector actions are necessary to allow switching between $Drive$ and $Halt$ components. In particular, connector $C1$ specifies that if $Drive$ reaches the end location $slowdown$, which is represented by interface locations $dl1 = "from" \wedge dl2 = "to" \wedge dl3 = "to"$, the control is passed to $Halt$ component entering its $slowdown$ location. This location is represented by interface location $hl1 = "from" \wedge hl2 = "to" \wedge hl3 = "to"$. The control is removed from component $Drive$ by assigning $dl1' = "" \wedge dl2' = "" \wedge dl3' = ""$.

*Definition 3 (serial composition):* A component $C$ composed in series from subcomponents $A$ and $B$; that is, $C = A \oplus B$, can be written as TLA$^+$module $M_C = \langle V_C, I_C, N_C \rangle$, where

- $V_C$ is a set of module's variables that includes all input and output variables of the subcomponents and location variables.
- $I_C$ is an initial predicate that is a disjunction of initial predicates of both submodules annotated with control flow information in the form of assertions on interface locations; that is, $I_C = (I_A \wedge L_B) \vee (I_B \wedge L_B)$, where $L_A$ or $L_B$ specifies that control can be assigned to component $A$ or $B$, respectively.
- $N_C$ is a next-state action predicate that is defined as a disjunction of next-state predicates of both submodules and all necessary connectors $C_i$; that is $N_C = N_A \vee N_B \vee C_i$.

MODULE $Engine$

EXTENDS $Integers$<br>
VARIABLES $brake,\ x,\ dx,\ clock$<br>
VARIABLES $hl1,\ hl2,\ hl3,\ dl1,\ dl2,\ dl3$

$drive \triangleq$ INSTANCE $Drive$ WITH $loc1 \leftarrow dl1,\ loc2 \leftarrow dl2,\ loc3 \leftarrow dl3$<br>
$halt \triangleq$ INSTANCE $Halt$ WITH $loc1 \leftarrow hl1,\ loc2 \leftarrow hl2,\ loc3 \leftarrow hl3$

$$\begin{aligned}
I1 \triangleq\ & \wedge dl1 = \text{“from”} \wedge dl2 = \text{“from”} \wedge dl3 = \text{“from”} \wedge drive!Init \\
& \wedge hl1 = \text{“”} \wedge hl2 = \text{“”} \wedge hl3 = \text{“”}
\end{aligned}$$

$$\begin{aligned}
I2 \triangleq\ & \wedge dl1 = \text{“”} \wedge dl2 = \text{“”} \wedge dl3 = \text{“”} \\
& \wedge hl1 = \text{“from”} \wedge hl2 = \text{“from”} \wedge hl3 = \text{“from”} \wedge halt!Init
\end{aligned}$$

$$Init \triangleq I1 \vee I2$$

$$L1 \triangleq halt!Idle \wedge drive!Next \wedge \text{UNCHANGED } \langle hl1,\ hl2,\ hl3 \rangle$$

$$\begin{aligned}
C1 \triangleq\ & \wedge dl1 = \text{“to”} \wedge dl2 = \text{“from”} \wedge dl3 = \text{“from”} \\
& \wedge hl1 = \text{“”} \wedge hl2 = \text{“”} \wedge hl3 = \text{“”} \\
& \wedge dl1' = \text{“”} \wedge dl2' = \text{“”} \wedge dl3' = \text{“”} \\
& \wedge hl1' = \text{“from”} \wedge hl2' = \text{“from”} \wedge hl3' = \text{“from”} \\
& \wedge \text{UNCHANGED } \langle brake,\ x,\ dx,\ clock \rangle
\end{aligned}$$

$$L2 \triangleq drive!Idle \wedge halt!Next \wedge \text{UNCHANGED } \langle dl1,\ dl2,\ dl3 \rangle$$

$$\begin{aligned}
C2 \triangleq\ & \wedge hl1 = \text{“to”} \wedge hl2 = \text{“from”} \wedge hl3 = \text{“from”} \\
& \wedge dl1 = \text{“”} \wedge dl2 = \text{“”} \wedge dl3 = \text{“”} \\
& \wedge hl1' = \text{“”} \wedge hl2' = \text{“”} \wedge hl3' = \text{“”} \\
& \wedge dl1' = \text{“from”} \wedge dl2' = \text{“from”} \wedge dl3' = \text{“from”} \\
& \wedge \text{UNCHANGED } \langle brake,\ x,\ dx,\ clock \rangle
\end{aligned}$$

$$Next \triangleq L1 \vee C1 \vee L2 \vee C2$$

Fig. 4. The TLA$^+$ specification of component $Engine$

## V. VERIFICATION USING TLC

In this section, a brief elaboration on results of verification experiments is presented. The TLC tool was used to check the basic properties of specifications composed in the style of Masaccio model.

Each component can be verified using TLC tool separately. Nevertheless, often a component depends on its environment and the environment specification needs to be supplied in order to get a meaningful results. For instance, component $Distance$ that computes a distance according to the actual velocity requires to provide a specification that sets boundaries on the behavior of velocity variable $dx$. Moreover, the dependency among the components can be circular. Therefore to verify a component, a suitable context needs to be provided. The approach used in this paper for verification of the components stems from the assume-guarantee principle that constraints the context of a component. This principle was studied in the frame of Masaccio formalism in [9]. The context does not need to be specified from scratch. Instead, existing specifications of components that form the environment of the component being verified can be turned into a context like specification. Moreover, this context specification can be proved as appropriate if it satisfies a refinement relation. In many cases, it can be checked automatically using TLC tool. Interface refinement is described in [3, p.163] as $LSpec \triangleq \exists \hat{h} : IR \wedge HSpec$, where $\hat{h}$ is a vector of free variables of $HSpec$ and $IR$ is a relation between variables of $\hat{h}$ and lower level variables of $\hat{l}$ of specification $LSpec$.

Verification of component $Engine$ required to compute the state space consisting of 816288 states. TLC completes this task roughly within 30 seconds on a computer with 1.66 GHz processor. It should be noted, that the specification shown in figure 4 was extended with $loop$ action for the sake of TLC verification procedure. The loop action is required to prevent TCL to complain on finding a deadlock state. It just loops forever if the end location of component $Engine$ is reached.

$$\begin{aligned}
loop \quad & \triangleq (hl3 = ”to” \vee dl3 = ”to”) \\
& \wedge \text{UNCHANGED } vars
\end{aligned}$$

The specification sent to TLC for verifying properties was as follows:

$$Spec == Init \wedge \Box[Next \vee loop]_{vars}$$

The verification of component $Near$ (see [4] for its specification) took much longer (approx. 15 minutes) and the state space searched was greater than 7 millions of distinct states. An issue lies in the use of integer variables for meassuring distance and actual speed of the train and the necessity to check whether the property holds for any combination of these values. The solution is to merge concrete values into significant intervals, i.e. for $dx$ there are two such intervals, in particular, $hispeed = (21..50)$ and $lospeed = (0..20)$. Also distance variable $x$ can be defined to be from a set of intervals, e.g. $extdist^+ = (\infty, 5000), fardist^+ = (5000, 1000), neardist^+ = (1000, 0)$.

## VI. Conclusion

In this paper, the overview of the method capable of formal specifying and verifying embedded control systems has been presented. The method is based on the TLA$^+$, which allows to produce clear and simple specifications because of its very expressive language. An accompanying tool, TLC model checker, can be employed to show that the specification exposes intended properties. This method was illustrated on a simple example in this paper. The semantics of specification can be defined in terms of Masaccio interpretation, including serial and parallel component compositions.

In addition to clarification of the basic facts on the method for writing TLA$^+$specifications under Masaccio interpretation, the several topics for future work were revealed during the work on this paper:

- Deeper understanding of the assume-guarantee refinement in the TLA$^+$specification framework is required and the proof that these specifications obey assume-guarantee principle as specified for Masaccio model should be given.
- Specification of hybrid systems as proposed by Masaccio was not addressed in the paper. As shown in [5], TLA$^+$is expressive enough to capture a large class of hybrid system specifications. The question is whether the verification can be adequately supported by the tools available for TLA$^+$.
- As expected and shown by the example, the state explosion is a problem in the case of verification of non-trivial systems. While TLA$^+$model-checker can explore several hundreds of millions of states, there is also possibility to apply state space reduction techniques. The TLC poses symmetry reduction mechanism [3, p.245] that can reduce significantly state space for design that contains multiple same or similar parts.

The formal model and the presented specification and verification method is suitable, in particular, for application to the domain of distributed time-triggered systems [10]. The intention is to integrate this method in a visual modeling framework [11] to enable automatic checking of properties of systems being visually modeled.

## References

[1] P. Cousot and R. Cousot, “Verification of embedded software: Problems and perspectives,” *Lecture Notes in Computer Science*, vol. 2211, pp. 97–114, 2001.

[2] E. Lee, *Advances in Computers*. Academic Press, 2002, ch. Embedded software.

[3] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley Professional, 2003.

[4] T. A. Henzinger, “Masaccio: A formal model for embedded components,” in *TCS ’00: Proceedings of the International Conference IFIP on Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics*. London, UK: Springer-Verlag, 2000, pp. 549–563.

[5] L. Lamport, “Hybrid systems in tla$^+$,” in *Hybrid Systems*, ser. Lecture Notes in Computer Science, vol. 736. Springer, 1992, pp. 77–102.

[6] M. Kaminski and Y. Yariv, “A real-time semantics of temporal logic of actions,” *Journal of Logic and Computation*, vol. 13, no. 6, pp. 921–937, 2001.

[7] L. Lamport, “Real-time model checking is really simple,” in *CHARME*, 2005, pp. 162–175.

[8] K.-K. Lau, V. Ukis, P. Velasco, and Z. Wang, “A component model for separation of control flow from computation in component-based systems,” *Electronic Notes in Theoretical Computer Science*, vol. 163, no. 1, pp. 57–69, September 2006.

[9] T. A. Henzinger, M. Minea, and V. Prabbu, *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, January 2001, vol. 2034/2001, ch. Assume-Guarantee Reasoning for Hierarchical Hybrid Systems, pp. 275–290.

[10] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, ser. The Springer International Series in Engineering and Computer Science. Springer Netherlands, 2002, vol. 395, ch. The Time-Triggered Architecture, pp. 285–297.

[11] M. Faugere, T. Bourbeau, R. de Simone, and S. Gerard, “Marte: Also an uml profile for modeling aadl applications,” in *ICECCS ’07: Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 359–364.

# Algorithm for Real Time Faces Detection in 3D Space.

Rauf Kh. Sadykhov
Belarusian State University of Informatics and Radioelectronics,
220125 Brovki str. 6 Minsk, Belarus
Email: rsadykhov@bsuir.by

Denis V. Lamovsky
Belarusian State University of Informatics and Radioelectronics,
220125 Brovki str. 6 Minsk, Belarus
Email: lamovsky@bsuir.by

***Abstract*—This paper presents the algorithm of stereo-faces detection in video sequences. A Stereo-face is a face of a man presented by set of images obtained from different points of view. Such data can be used for faces structure estimation. Our algorithm is based on computationally effective method of face detection in mono image. Information about face positions is then combined using sparse stereo matching algorithm. Sparse means that stereo correspondence is estimated not for all scene points but only for points of interest. This allows obtaining the low computation cost of algorithm. We use few criteria to estimate correspondence. These are: epipolar constraint, size correspondence, 3D region of interest constraint and histogram correspondence.**

**Object distance estimation method that does not use projective transformations of stereo planes is also considered.**

## I. Introduction

3D FACE modeling is widely used nowadays in data processing systems. These models are applied in such fields as man-computer interface, multimedia applications, biometric identification in surveillance and access control systems. Three-dimensional face recognition became more popular in recent time [1], [2], [3]. Its methods promise to achieve better accuracy in comparison with the more traditional 2D face recognition methods. It is rapidly developed area in computer vision.

Special equipment is usually used to obtain 3D models. It can consist of several video cameras, structured light source, special laser scanners and other. When accuracy of the model is not critical 3D models can be obtained from static images or frames of video sequence [4]. Building such coarse models sometimes requires user assistance [5]. Mentioned approaches allow estimating three-dimensional face structure with appropriate quality but have disadvantages. They can't be used when it is necessary to obtain such models in real time, in uncontrolled conditions and when user assistance is unavailable.

Person identification systems which use 3D face models as biometric feature require high speed of the model estimation and minimum requirements to the object position. Existing systems can not provide such requirements. A lot of cameras of surveillance systems are used in present time. Such equipment can provide obtaining frames with high resolution. They can be used for 3D face estimation and can be done in several ways. The first approach is based on mono view image processing, the second are multi view (stereo) images processing. In the first case frontal face position is strongly required for acceptable model accuracy. Such models can not accurately represent real parameters of the face. A more accurate face model can be obtained using images from different points of view. Stereo reconstruction methods can be used in this case for additional information estimation. Stereo base face detection methods [6] require additional efforts for depth estimation. It also usually need that cameras have parallel optical axes. In other case it will be necessary to perform transformation of the images to a standard parallel stereo view [7].

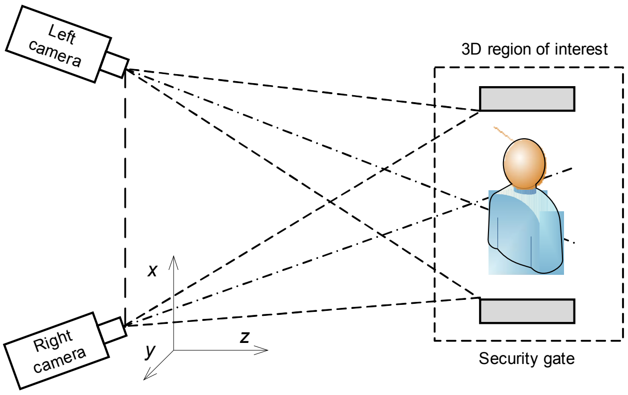


Fig. 1. 3D face processing in security applications

Object detection is usually one of the stages of image processing. We speak about face detection in the case of person identification by portrait. In this paper we present the algorithm of face detection and matching in frames of stereo images sequence. Our goal is to detect and localize stereo-faces of people moving in the field of view of several video cameras (see fig. 1). Stereo-face here is a set of face images obtained from different views. Main requirements are the next: number of faces from 1 to 5; face size not less than 100*100 pixels; and frame size is 1024*768 pixels.

## II. Camera Calibration

Camera calibration in stereo vision classically means computation of relative positions of the camera optical centers. It is not necessary for current task and it is acceptable to compute fundamental matrix that represents epipolar geometry of

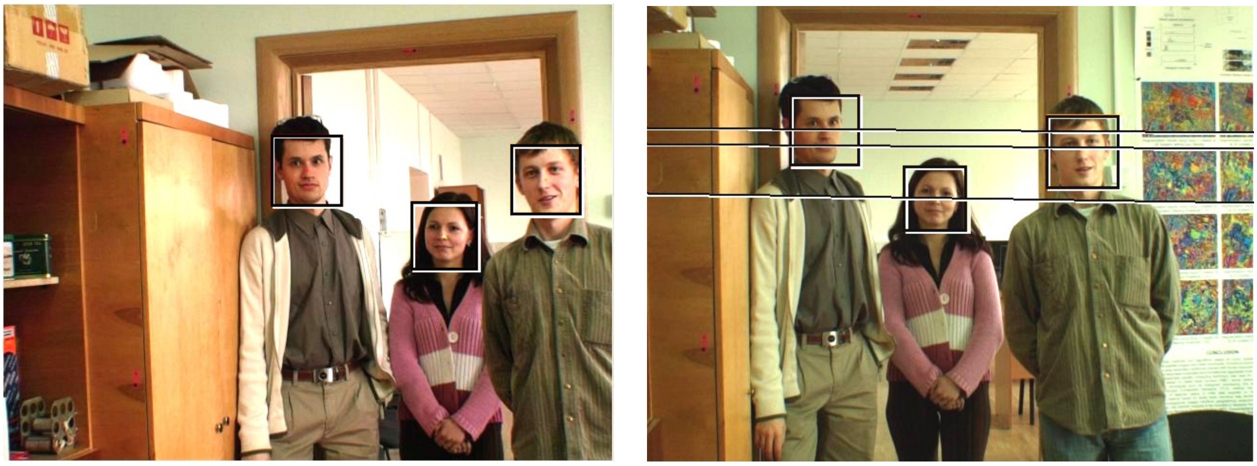


Fig. 2. Stereo frame with detected faces and corresponding epipolar lines.

camera views. The fundamental matrix allows calculating so called epipolar line in one frame that corresponds to given point in another frame. Epipolar geometry greatly reduces search space for points correspondence estimation. We used RANSAC method [8] that is based on eight points algorithm [9] for fundamental matrix estimation.

User has to mark at least eight pairs of correspondent points for calibration. Obtained matrix can be applied for epipolar line estimation by using next relation:

$$l = Fx, \tag{1}$$

where $x$ – given point, $F$ – fundamental matrix and $l$ – sought value.

## III. Stereo-Faces Detection

We use sparse stereo matching algorithm for stereo-faces obtaining. It works with small amount of scene points that leads to fast processing speed. In classical scheme of stereo reconstruction correspondence is estimated for each point of stereo frames. Our algorithm processes only points of interests. These points correspond to human faces found at each view. We use cascade of weak Haar classifiers [10] for face detection in each frame. Fig 1 shows stereo pair example with detected faces and corresponding epilines.

### *A. Matching Algorithm*

Applied approach for face corresponding estimation is based on geometric characteristics analysis, spatial face position and visual correlation. Faces found in each view are characterized by their geometric characteristics and pattern. The algorithm calculates correspondence degree for objects from two frames based on these characteristics. The goal of this procedure is selection of most probable pairs and rejection of faces without pair.

Pair correspondence is defined by multiplication of the set of coefficients. Each coefficient corresponds to some feature correlation:

- K1 – represents the faces positions accuracy in compliance with epipolar geometry;
- K2 – represents face sizes correspondence;
- K3 – represents histogram correspondence of the face areas.

All calculation are performed in the coordinate system of one frame from stereo pair. Algorithm handles only pairs that comply with epipolar constraint. In other words, pair candidates for the face from one view have to lie on correspondent epiline at another view. In fig.2 each man's face in first image has two candidate faces in second image, but woman's face has only one. This constraint allows greatly reducing the amount of false pairs. Then size and position coefficients are calculated.

Coefficient K1 is defined as ratio between distance from face area centre to correspondent epipolar line and minimal face size in pair. It possesses the value 1 when the center of the face coincides with the line and is less than 1 otherwise. Coefficient K2 is defined as ratio between smaller and bigger area sizes. It also is less or equal to 1.

It is necessary to estimate face position in the 3D space for coefficient K3 calculation.

### *B. Histogram matching.*

Coefficients K1 and K2 allow rejecting big amount of false pairs. They are not enough because are based on geometrical features that can be obtained with essential error. That is why histogram matching based coefficient K3 is

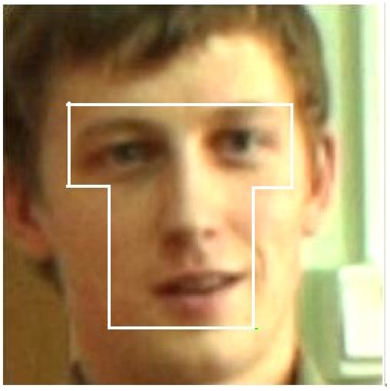
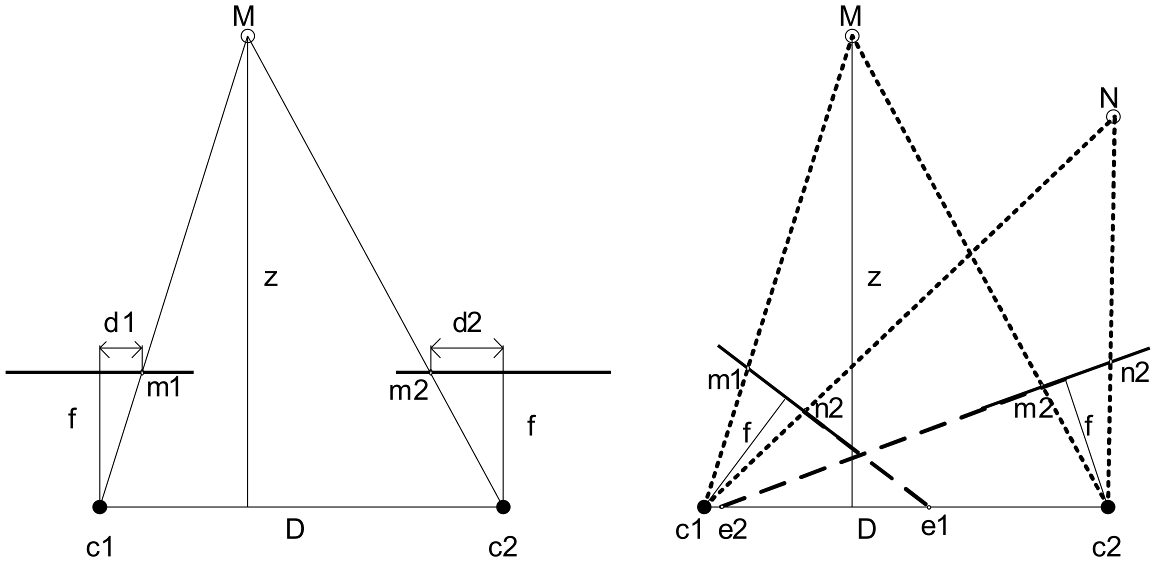


Fig. 3. Histogram area



a b

Fig. 4. Stereo geometry a) simple, b) with arbitrary cameras positions.

computed for the rest small amount of possible pairs. Histogram matching is used because of its relative computational simplicity in comparison with other metrics.

Histogram is built not for all face area to avoid influence of background, hair, clothes pieces that can be not presented at both views. Only central part of face area is used where eyes, nose and mouth are apparently situated (see fig. 3). Median filter and area normalization are used to bring histogram to common form.

Sum of Absolute Differences (SAD) and Sum of Squared Differences were used for histogram matching. The SAD applied to three-cannel histograms shows the best result. Error is calculated as square of Euclidian distance between each RGB component's errors. Coefficient K4 then is estimated as:

$$K3 = 1 - histERR\,, \tag{2}$$

where *histERR* is histogram correspondence error.

Hence maximization of the product *K1*K2*K3* gives us the most probable pairs. Pairs formed by faces without right correspondence are ignored by threshold. Such pairs can appear because only one face view can be visible for stereo cameras. The threshold is applied to compound coefficient (the product). Its value was estimated empirically and is equal to 0,3.

## IV. Face 3D Position Estimation.

It is known from stereomerty that distance to object can be estimated by using triangulation. Fig 3a. shows simple stereo system geometry. The indications are: c1 and c2 – camera optical centers, M – point of the scene, m1 and m2 – scene point projections, *f*–focal length of optical system and *D*–distance between optical centers (so called base line). Unknown distance *z* can be found by using next equation:

$$\frac{z}{D} = \frac{f}{d1 + d2}\,, \tag{3}$$

where sum *d1+d2* is disparity of projections' positions.

Real stereo systems rarely have such ideal configuration. Fig.4b represents the geometry of a stereo system with arbitrary cameras position and orientation. Image planes in such system are not parallel to base line. It is necessary to perform projective transformation of each plane in order to use eq. 3 for distance estimation. It leads to additional computational cost.

Optical centers' projections can not be used as reference points for disparity computation in this case. It is seen in fig. 4b that disparity for space points M and N relative to image centers are equal. But these points lie at different distance from base line. It is necessary to choose reference points for disparity computation that allow estimating distance independently from camera configuration. We use epipolar centers as such points. These centers are points where all epipolar lines of respective view come together. The centers are indicated as e1 and e2 at fig. 4b. These points can be calculated by using next equations:

$$\begin{aligned} e_1 F &= 0, \\ e_2 F^T &= 0, \end{aligned} \tag{4}$$

where *F* is fundamental matrix.

Disparity computed relative to epicenters can not be used for real distance calculation but allows estimating its value. Disparity evaluations can have large absolute values. Especially in the case when image plane has very small angle relative to base line. Such values are not informative. We consider the crossing point of cameras' optical centers as zero point (Z=0) to avoid large disparity values. Base disparity *d*0 is calculated for this point. Distance to the object is estimated as difference between the point and base disparities. For example absolute disparities *dm* and *dn* for points M and N at fig. 2b can be estimated as:

$$dm = dist(m1,e1)+dist(m2,e2), \tag{4}$$
$$dn = dist(n1,e1)+dist(n2,e2),$$

where *dist* is distance function between two points. Distance (or depth) evaluation can be computed as *dm-d*0 and *dn-d*0 respectively.

Hence three-dimensional face position accuracy in compliance with 3D region of interest can be computed as:

$$K_{ROI} = 1 - \frac{|d_{face} - d_0|}{\Delta d}, \tag{5}$$

where $\Delta d$ is disparity scattering determined by 3D region of interest.

## V. Position estimation discussion

It seems that the coefficient $K_{ROI}$ can to be used as additional weight for pair correspondence estimation. It means that it can be added to the complex coefficient ($K1*K2*K3*K_{ROI}$). Figure 5 illustrates this idea.

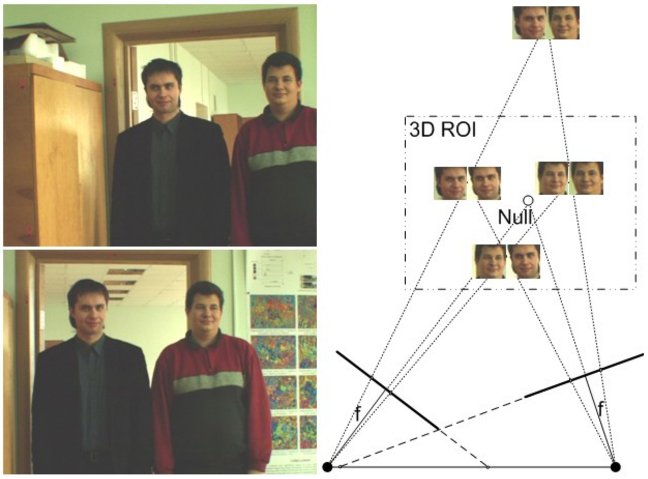


Fig. 5. Peoples stay close to zero point.

Men's stay quite close to zero point inside 3D ROI. False face pairs are located behind and in front of true pair's positions. It means that difference between true face disparity and zero point disparity is less than the same value in false face case. False face means that it is formed by false pair. Coefficient $K_{ROI}$ (5) will be mach less for such pairs than for correct pairs. It allows greatly increasing of the right correspondence rate.

It was good idea, but its application meat problems. Some cases exist then this condition does not help. Figure 6 represent the example then people stay close (at the same distance) to each other but relatively far from zero point.

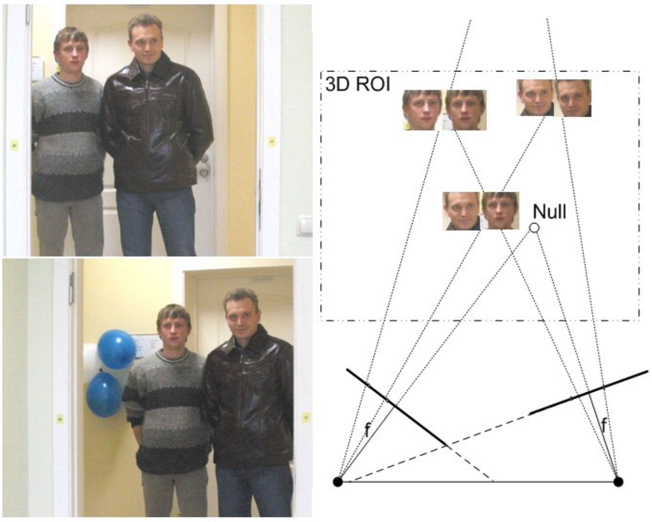


Fig. 6. Peoples stay far behind zero point.

Here false pair has greater weight because it lies closer to zero point. Right and face pairs positions are unpredictable. It is impossible to say even that they always positioned at the same cornets of the quadrilateral (fig. 7).

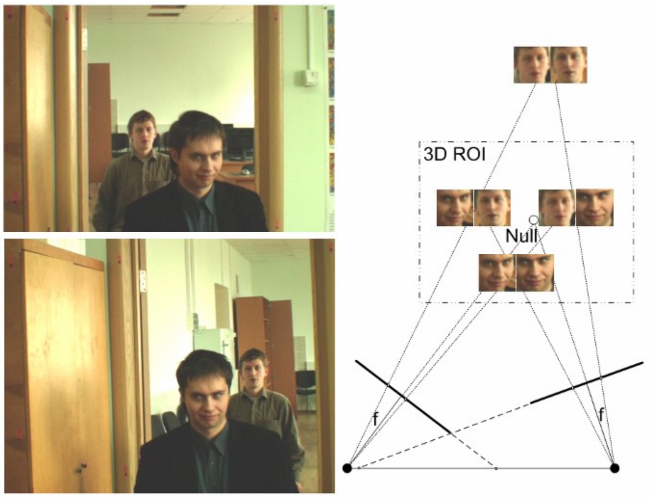


Fig. 7. Peoples stay at the different sides from zero point.

The obstacle problem also leads to unpredictable values of the coefficients for correct and false pairs. We considered different cases of occlusion then a face can close one ore more other faces on different views. We skip its detailed description.

As the result we can say that coefficient $K_{ROI}$ can't be used for correspondence estimation as it was described. But it can be applied for face ranging. It can be usable to process faces with predictable parameters. For example face identification method can be configured to operate with appropriate size faces. This size must correspond to visible face size then corresponding man stay at the zero point distance.

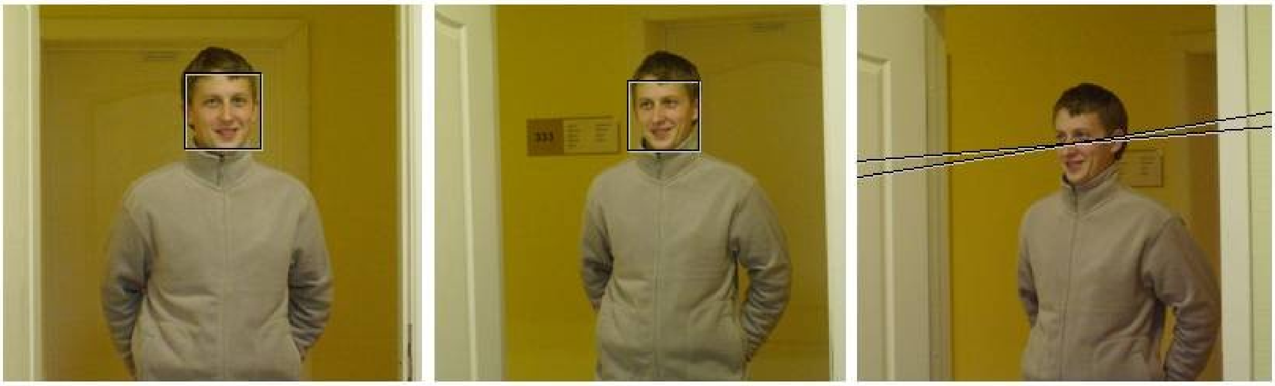


Fig. 8. Missed views estimation based on epipolar geometry.

## VI. Missed Views Estimation

Described algorithm can be used in a system this two and more cameras. The situation then face was not found at all vies is possible. It may be causes by to big angle of the head rotation, occlusions, mistakes and so on. If face was detected at least at two frames its position can be computed at other views based on epipolar geometry.

Position of the missed face view is defined by crossing two ore more epipolar lines which correspond to known views. For example, face was detected at first and second frames (Fig. 8) and they were matched by presented algorithm. It is necessary to compute epipolar line at third frame that correspond to the face at first frame and also epipolar line that correspond to the face at second frame. Face must be positioned on each epipolar line. That's why the crossing is the result (Fig 8. right image).

## VII. Results

The algorithm has been tested with images obtained by stereo device that includes two high resolution cameras. Stereo frames are color images with resolution 1024*768. It was considered that maximum face size is 100*100 pixels. Test set includes 200 stereo pairs and Table 1 represents test results.

The matching accuracy came to 93.5%. Time costs for processing is about 50 ms. per stereo frame. Main part of this processing time employs face detection module. Hence presented algorithm allows face views matching with high processing speed.

Table I.
Algorithm Test Results

| Parameter | Numerical value | Percentage |
|---|---|---|
| Number of faces in left view | 336 | - |
| Number of faces in right view | 372 | - |
| Detected faces (left) | 326 | 97% |
| Detected faces (right) | 366 | 98% |
| Total amount of correct pairs | 306 | - |
| Correctly matched pairs | 286 | **93,5%** |
| False rejections by threshold | 4 | 1,3% |
| False matched pairs | 2 | 0,65% |
| Faces without pair (left) | 30 | 13% |
| Faces without pair (right) | 61 | |

## VIII. Future Work

Described algorithm was developed to be used in biometric identification applications which operate in real time. Several ways to use the results of this algorithm for this purpose exist.

First of all stereo face can be used for best view estimation. 2D face recognition algorithms mostly based on frontal face processing. Information about relative positions of eyes, nose and other feature point can be used for appropriate image selection. Needed information can be obtained using methods of stereo analysis.

From other hand good frontal view can be not represented at frames. It is very probable situation especially if only two cameras are used. In this case stereo face can be used to develop frontal face view.

The most preferable case of stereo face usage is 3D model estimation (Fig. 9). Such techniques are still very computationally expensive and require a lot of efforts to be launched in real time.

## IX. Conclusion

Developed algorithm allows obtaining 3D position of stereo-faces with high computational efficiency. It is based on sparse stereo matching of objects of interest. Fast speed processing was also reached by using distance estimation procedure without projective transformation. It can be used in real time surveillance systems as base for 3D face identification.

## References

[1] S. Gupta, M. K. Markey, and A. C. Bovik, "Advances and challenges in 3D and 2D+3D human face recognition," in *Pattern recognition in biology*, M. S. Corrigan, Ed., Nova Science Punlisher Inc., 2007, pp. 63-103.

[2] L. Akarun, B. Gokberk, and A. A. Salah, "3D face recognition for biometric applications," in *Proc. 13th European Signal Processing Conference,* Antalya, Turkey, 2005

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Three-dimensional face recognition," *International Journal of Computer Vision*, vol. 64, no. 1, 2005, pp. 5-30.

[4] V. Blanz, and T. Vetter, "A Morphable model for the synthesis of 3D faces," in *Proc. 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 187–194.

[5] Z. Liu, Z. Zhang, C. Jacobs, and M Cohen, "Rapid modeling of animated faces from video" unpublished.

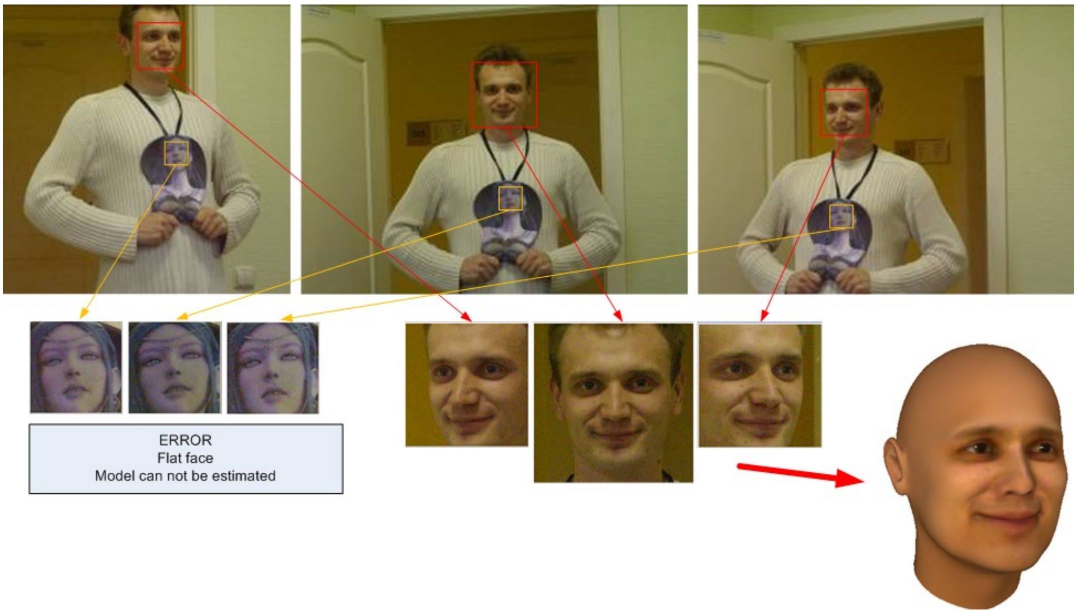


Fig. 9. 3D face model estimation using stereo faces.

[6] C. Park, and J. Paik, "Face recognition using optimized 3D information from stereo images," *Image Analysis and Recognition*, 2005, pp. 1048-1056.

[7] R. Niese, A. Al-Hamadi, and B. Michaelis, “A novel method for 3D face detection and normalization,” *Journal of Multimedia, Academy Publisher* vol. 2, no. 5, 2007, pp. 19-26.

[8] A. J. Lacey, N. Pinitkarn, and N. A. Thacker, “An evaluation of the performance of RANSAC algorithms for stereo camera calibration,” in *Proc. 11th British Machine Vision Conference*, Bristol, 2000.

[9] I. H. Richard, “In defense of the eight-point algorithm,” *IEEE Trans. Pattern Analysis Machine. Intelligent,* vol. 19, no. 6, 1997, pp. 580-593.

[10] P. Viola, and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, 2004, pp. 137-154.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 733 – 737

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Reliability of Malfunction Tolerance

Igor Schagaev
Dept. of Computing
London Metropolitan University
166-220 Holloway Road, London N7 8DB, United Kingdom
Email: i.schagaev@londonmet.ac.uk

*Abstract*—**Generalized algorithm of fault tolerance is presented, using *time*, *structural* and *information* redundancy types. It is shown that algorithm of fault tolerance might be implemented using hardware and software. It is also shown that for the design of efficient fault tolerant system elements must be malfunction tolerant. The advantage of element malfunction tolerance is proven in reliability terms. Reliability analysis of a fault tolerant system is performed with deliberate separation of malfunction and permanent faults. Special function of reliability gain is introduced and used to model system reliability. Maximum reliability of fault tolerant system is achievable with less than a duplication system and depends on malfunction/permanent fault ratio and coverage of faults. System level of fault tolerance prerogative is the reconfiguration from permanent faults.**

## I. Introduction

THE world of applications for fault tolerant systems is expanding beyond our imagination. Areas where we need them most are called embedded systems. The principal features of on-board systems as well as embedded safety critical systems are Fault Tolerance (FT) and Real Time (RT) response. They both are reflected in the main system components: hardware (HW) and system software (SSW). The process of FT system design consists of two mutually dependent sequences of HW and SSW development, using different redundancy types [1]. The design of the FT system assumes that the required specification is already known, including faults that system should tolerate [2],[14],[25].

FT system design required objectives are: performance, reliability and low power consumption (the latter especially for spaceborne computers). All of them are achievable in combination by means of a careful and balanced introduction and monitoring of hardware and system software features. Achieving reliability assumes either using of multiple unreliable components [3], or higher reliability components, or application of various types of internal redundancy to maximize efficiency of the solution [4],[5].

The term *fault tolerant system* needs a rigorous definition; at the same time it must be differentiated from such terms as *graceful degradation* and *fail-stop systems* . Avizienis [6-10], Laprie [9,11-12] and Siewiorek [14] proposed that a system be called *fault tolerant* (FTS) if it recovers itself to full performance or at least continues with sufficient specified functions and required features.

The *gracefully degradable system* (GDS) is the system that can recover itself and continue functioning in degraded mode after occurrence of a fault [12] . In turn, if a system can stop itself correctly once a fault has been detected in acceptable state it is called a *fail-stop system* (FSS) [13].

The most critical reliability requirements for on-board applications are MTTF (Mean Time To Failure) 10-25 years and system availability not less than 0.99 over the whole life cycle of the system, typical for aircraft, satellites, gas pipelines, etc.

## II. Generalized Algorithm of Fault Tolerance

Several authors [2],[5] proposed to consider fault tolerance as a process of several steps for: proving that the appeared fault did not damage hardware, determination of type of fault (malfunction or permanent), checking the consistency of software state, detecting of correct states and further recovery from the correct state.

This process was called a Generalized Algorithm of Fault Tolerance (GAFT). The *information* , *time* and *structural* types of system redundancy enable to complete these steps of the process of fault tolerance. Combination of GAFT and possible redundancy types forms a matrix shown in Table 1. All steps of GAFT are implemented by SSW and HW. Steps {A,…,G} vs. redundancy types form a framework for the classification, design and analysis of implemented FT computer systems. According to Table 1, a system is called *fault tolerant* if it implements GAFT, i.e. every row (line) was visited and the algorithm is completed. There is no doubt that the algorithm can be completed differently, using different redundancy types at each step. This shows that fault tolerant systems may differ in:

- The time taken to implement various steps
- Types of redundancy used
- Types of faults which might be tolerated.

The taxonomy of Table 1 might be used for comparison of various design solutions of FT systems. The other interesting property of this taxonomy is an evaluation of efficiency of redundancy in implementation of various steps of GAFT. Therefore, this taxonomy might provide analytic evaluation and assist in selection of the most efficient solution for the

This work has been done as a part in the ONBASS (On Board Active Safety System) Project No. 160045, funded by the EC.

implementation of on-board system from the alternative approaches.

TABLE I.
GAFT VS. REDUNDANCY TYPES

| Steps | Redundancytypes | | | | | |
|---|---|---|---|---|---|---|
| | hw | | | sw | | |
| | i | s | t | i | s | t |
| A.Provetheabsenceof fault, ELSE | | | | | | |
| B.Determinetypeof fault | | | | | | |
| C.If faultis permanentTHEN | | | | | | |
| D.Reconfigurehardware | | | | | | |
| E.Proveconsistencyof softwareELSE | | | | | | |
| F.Locatefaultystates | | | | | | |
| G.Recoversoftware | | | | | | |

Description of a fault tolerant system proposed initially for on-board systems [5] is more rigorous. It applies an algorithmic definition of the feature of *fault tolerance* using GAFT. In other words, GAFT introduces FT not as a feature but as *a process*.

We consider a system as *fault tolerant if and only if* it implements GAFT transparently for an application. On-board system's main function is the implementation of control algorithms; we call it "process one" or $P_1$ [15]. Thus the functional definition of fault tolerance for control computers is the following: if the system provides for an application ("process one") full functionality and can transparently for the application recover itself from a predefined set of faults, this system will be *fault tolerant* .

The transparency of fault tolerance for the applications (for on board computers) means that GAFT is completed within a defined time (between sequential data outputs or inputs; in practice it is about 10-125 milliseconds). In short, the system is fault tolerant if it provides failure-free-mode for the implementation of the process $P_1$ . Again, it is assumed that the performed recovery is transparent for the application.

## III. TIME REDUNDANCY AND FAULT TOLERANCE

The GAFT implementation varies in terms of redundancy types used and, therefore, the time to complete. In terms of time slot which is required to implement FT of the system, different levels of granularity of SW might be used: instruction, procedure, module, task and system, as presented in Fig. 1.

At the *instruction level* scheme, SSW assumes an elimination of a fault appearing and its influence within the instruction execution. For example, the triplicate memory voter masks the faults of one memory element and therefore, this fault is tolerated at the *instruction level* .

In turn, the techniques such as microinstruction or instruction repetition can be considered as implementation of instruction level FT for the processor. From the system point of view an implementation of fault tolerance at the *instruction level* has a serious and undisputable advantage: fault checking, detection and recovery can be completely transparent to the system software.

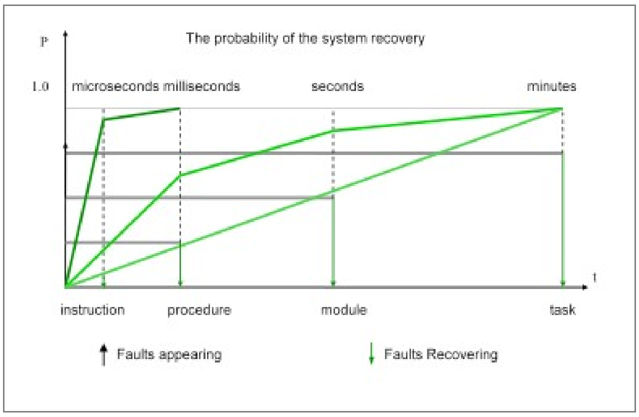


Fig. 1 Levels of implementation of fault tolerance

At the *procedure level* an implementation of fault tolerance assumes that a fault and its influence is tolerated and eliminated within the scope of a procedure execution. For example, the techniques such as *recovery blocks* (wrongly known as recovery points) [16],[17],[18],[19] can be considered as procedure- oriented scheme. This approach requires system software support (compiler level and run-time system) and might be efficient if and only structural programming approach is used (known after [20]).

The analysis of efficiency of procedure level approach together with aspects of implementation was done in [21]-[23]. There is no doubt that the state of the program must be conserved/re-generated to recover from a hardware fault. For this scheme the volume of the information required to restore the system after fault is much larger (and has significant software overheads).

For the *module level* scheme a module tolerates influence of a fault. For example, module restart or run of a simplified alternative module [10] might be used. Module recovery reduces the probability to restart the task or the whole system. The same comments apply for the procedure scheme.

At the *Task level* a fault and its influence is eliminated by restart of a task after reconfiguration of hardware.

Initialization of this scheme of GAFT implementation might also use time redundancy available within task execution. Clearly this option has the highest software, state space and time overheads for implementing fault tolerance. This level requires system software (operating system) support for HW fault tolerance. Details of the SSW features to support hardware fault tolerance for parallel FT systems are discussed in [24].

*System level* of fault tolerance assumes automatic or manual (when the control process allows it) restart of the system after fault and performing of control function after full restart of the system. Automatic restart has to be supported by system software using automatic hardware testing and software restart schemes.

## IV. PERFORMANCE OF FT SYSTEMS

The time impact of implementing GAFT at each of these levels is different, as well as is the delays caused by their use. The current embedded system practice indicates the following order of magnitudes for timing:

- microseconds for the instruction level
- milliseconds for the procedure level
- hundreds of milliseconds for the module level
- tens of seconds at the task level
- ten of minutes to hours at the system level.

The various schemes of GAFT implementation have different overheads, capabilities for tolerating of various classes of faults, power consumption overheads and affiliated system costs. For the instruction level scheme the concrete hardware support is required. For example, duplicate or triplicate hardware modules could result in serious overhead in power consumption and size. Some better solution was recently found as well (http://www.itacsltd.com).

However, not all hardware sub-systems need necessarily be designed in this way. For example, a cost benefit analysis (where 'cost' infers financial, power consumption, chip area, time delays, etc.) might indicate that it is worthwhile to have the processor and system RAM fault tolerance, achieving fault tolerance of the rest of the system using procedure or module schemes.

The *procedure* and *module level* schemes require much less hardware support but, of course, have a larger timing and software-coding overhead. For the *module* and the *task level* schemes with no extra hardware, system software support for fault tolerance is actually incomplete, as any *permanent* hardware fault would cause the system stop and loss of the control application. If the system requires reboot to recover then it is resilient to *malfunctions* (transient faults), not the permanent faults.

Depending on the level employed in the implementation of fault tolerance, systems differ in time required of achieving it (see the thick lines in Fig. 1). Note that different fault types might be tolerated by different schemes applied in combination; in other words, levels of implementation of fault tolerance are not mutually exclusive.

A good fault tolerant system tolerates the vast majority of malfunctions within the *instruction* execution making them invisible in terms of other instructions (and software) . The malfunctions with longer time range might be detected and recovered differently, for example, at the procedural or task level.

The complexity of GAFT implementation also differs in the types of fault that have to be tolerated. Even knowing that the ratio of malfunctions is the order of magnitude higher than permanent faults, it is necessary to implement the special schemes for re-configurability and recoverability of the hardware to eliminate an impact of *permanent faults* on the system.

At the other extreme, the system could tolerate the vast majority of *malfunctions* at the task level using SW. The operating system support or even user support also might be used to tolerate hardware faults. From the user's viewpoint even 'WINTEL' systems (Windows/Inter based) might be considered as fault tolerant as long as scheduled application was completed and results delivered on time. In this case, the fault tolerance was achieved by means of maintenance of hardware and involvement of system engineers to fix the fault 'in time'.

Most WINTEL systems assume system reboot and restart of the applications is an acceptable way of operation. Practical experience of real-time systems confirms that this is not the case, at least for systems that must run continuously for more than a few hours.

## V. Reliability Evaluation for Fault Tolerance

From the classic reliability point of view any extra redundancy of hardware reduces the absolute reliability of the system [26]. At the same time, a reliability of the system might be increased if introduced redundancy is able to tolerate faults of the main part of the system by means of GAFT and itself contributes less in the system fault ratio.

Thus, part of the problem – the decreasing reliability of the system caused by hardware redundancy at some point becomes part of solution. GAFT implementation breaks down to two processes for checking (P2) and recovery (P3), while process (P1) is application execution [15]. Clearly, there are design tradeoffs to be made to achieve the optimum operational reliability and redundancy types used.

Reliability analysis [27] introduces a reliability value for each element (hardware redundancy) and assumes a Poisson failure rate $\lambda$ . This makes possible to calculate the overall reliability as a function of time for the whole system, if the structure of the system is known.

Hardware redundancy used at the various steps of GAFT degrades in reliability over time; thus achievable performance and reliability and their degradation within life cycle of the RT system are dependent. Therefore, an analysis of the surface shape and evaluation of performance and reliability degradation caused by the redundancies used should be performed for every fault tolerant system. Fig.2 presents qualitatively a slope where a fault tolerant system should be located, between the plane of requirements and curves of reliability and performance degradation.

Furthermore, the introduction of the cost to implement each proposed solution makes it possible to summarize the system overheads required to implement fault tolerance. There is no doubt, a quantitative evaluation of reliability, performance and cost overheads within one framework might be extremely efficient for justification of the design decisions and comparison of different approaches in implementation of fault tolerance.

There is a correspondence between reliability of FT systems and steps of GAFT related to the malfunction tolerance. The next sections analyze this correspondence.

### *A.Impact on Reliability of Malfunction Tolerance*

Let's denote a rate of permanent fault for the system without redundancy as $\boldsymbol{\lambda}_{pfl}$ , the malfunction rate as $\boldsymbol{k}\,\boldsymbol{\lambda}_{pfl}$ . For modern technologies $\boldsymbol{k}$ varies from $10^{3}$ to $10^{5}$ (the latter applies to aeronautics). The probability of operation without permanent fault within time gap [0,T) and mean time to failure are determined by formulas (1).

$$P_{1}(t) = e^{-((1+k)\,\lambda\, pf1)t}$$

$$\mathrm{MTTF}_1 = 1/(1+k)\lambda_{pfl} \qquad (1)$$

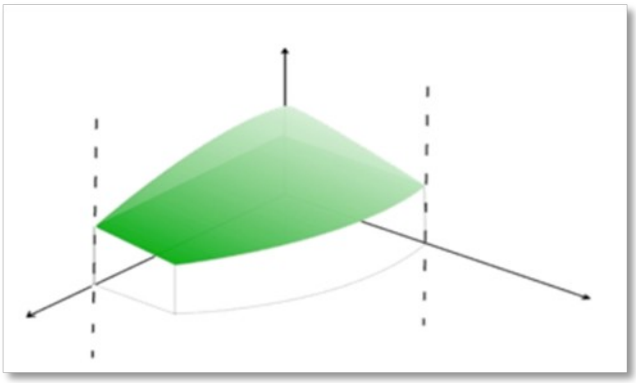


Fig. 2 Performance/reliability dependence

Fig. 3 presents a reliability degradation of the device without implementation of the fault detection and fault recovery). Introduction of the checking of hardware condition requires a hardware support.

In reliability terms, extra hardware has an extra rate of permanent and intermittent faults d λ pf1 and d λ if1 respectively, and reliability of the system with an implemented checking process is equal

$$P_{1+checking}(t) = e^{-((1+d)\ \lambda\ pf1+ (1+d)\ \lambda\ if1)t} \qquad (2)$$

where ***d*** is a share of the hardware redundancy required to implement the checking process. Redundancy for checking (detection) d varies in different hardware schemes. The maximum redundancy to provide checking is 100%, or d=1, when duplication of the hardware is used to compare outputs.

Maximum power of fault detection is a privilege of the duplication approach. Assume the same coefficient k of malfunction/permanent fault ratio

$$P_{1+checking}(t) = e^{-((1+d)+k)\lambda\ pf^{*}t}$$

$$MTTF_2 = 1/(1+d)+k)\lambda_{pf1} \qquad (3)$$

Introduction of checking schemes makes the only difference in system reliability analysis by a guarantee that the appeared hardware fault (assuming the chosen fault type) is detectable. In other words, checking justifies reliability by transition of an observer from a random world into the world of conditional probabilities. Conditional probability describes a reliability of the system provided that fault of a chosen class did not happen.

So far, we assumed that our system has hardware to detect chosen and representative class of faults and that faults do not have latency period (a natural assumption for processors, cache memory and immune systems).

In contrast to detection, a recoverability of the system after intermittent faults requires more efforts and redundancy use, let's denote it r. Then the redundancy of hardware descends even more – the lowest curve of Fig 3.

$$P_3 = e^{-(1+d+r)(\lambda\ pf1)t} \qquad (4)$$

$$\mathrm{MTTF}_3 = 1/(1+d+r)(1+k)\lambda_{pf1}t$$

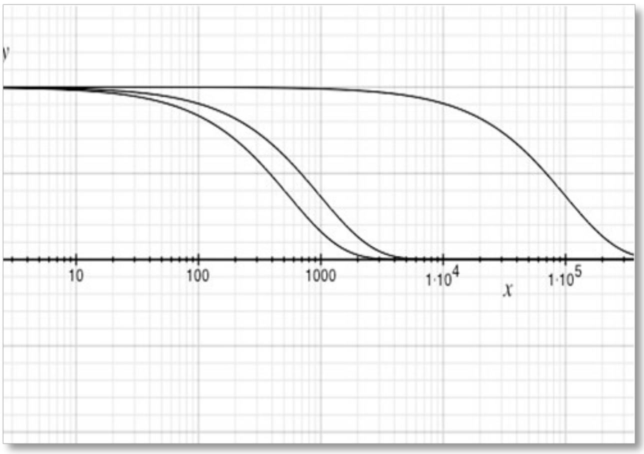


Fig. 3 R(t) with malfunction and checking

On the other hand, reliability might be increased, as long as malfunctions are tolerated. In the equations (4), a coefficient k of malfunction impact might be reduced, say, α times. Then

$$P_{1+checkingandrecovery} = e^{-(1+d+r)(1+\alpha k)(\lambda pf1)t} \qquad (5)$$

$$\mathrm{MTTF}_4 = 1/(1+d+r)(1+\alpha\, k)\lambda_{pf1}t$$

The role of α defines a reliability gain that might be achieved by elimination of malfunctions. It is worth to analyze α a bit more.

### *B. Success Function*

Let us define a success function of the malfunction reductions as SF. It is a well known fact that a double system covers all possible faults by comparison and while checking of the element is implemented it can recover from malfunctions [21]. Thus, a form of SF is defined by two values: initial and maximal. When no redundancy is used (x=0), a probability of recovery from malfunction is zero, SF→0. (Compare: if you don't pay for people healthcare then people will die from any disease…). In turn, using the known fact that 100% redundancy guarantees full success of system operation (including fault detection and complete recovery) one is able to write that SF →1.

A function SF=x*e1-x satisfies both initial conditions. Denoting coefficient alpha α as malfunctions reduction leads to:

$$\alpha = 1- c^{*}(x^{*}e^{\ 1-x}) \qquad \mathbf{(6)}$$

where ***c*** is a coverage of faults and directly depends on redundancy spent on detection c = f(d,λ) - the more we know the better, but at the cost.

Compare finally MTTF2 and MTTF1 (for the systems with malfunction elimination and without, respectively) calling it efficiency. Y axis denotes times of improvement, while X axis presents amount of redundancy involved:

$$E = (1+k)/[(1+d+r)(1+(1-c^{*}x^{*}e^{1-x})k)] \qquad (7)$$

Figure 4 shows that higher coverage of fault and ability to tolerate malfunctions makes an element almost 3 times more efficient in comparison with a standard element. The gain will be much higher when a real malfunction to permanent fault ratio (100:1 or 1000:1) is applied.

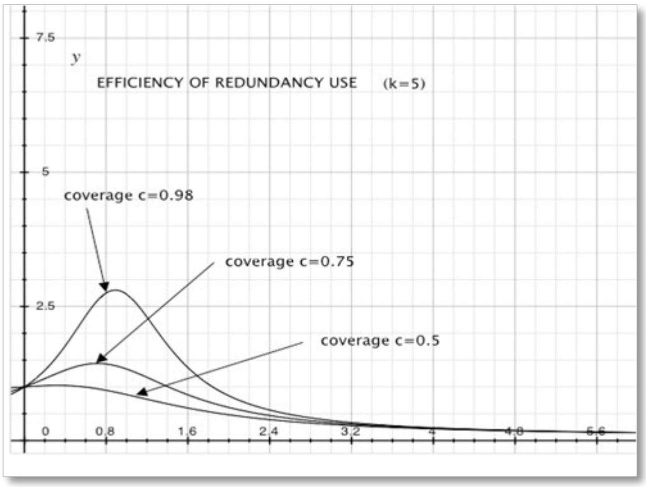


Fig. 4 Efficiency of redundancy

The estimation presented in Fig 4 does not separate use of redundancy between checking and recovery processes. This is a subject of further research; but the key outcomes are obvious and summarized below.

## VI. Conclusion

Design of fault tolerant systems should be revisited: malfunctions must be tolerated at the element level, leaving permanent fault handling to the system level. There is an optimum in redundancy level spent on fault tolerance; this optimum is achieved with less than 100% redundancy. Efficiency of fault tolerance implementation in reliability terms depends on coverage of faults.

## Acknowledgement

The author thanks Prof. Janusz Zalewski and Prof. Andrew Kornecki for encouragement, support and critique in the preparation of the paper.

## References

[1] Schagaev I., Using information redundancy for program rollback, *Automatic and Remote Control* , pp. 1009-1017, July 1986.

[2] Sogomonyan E., Schagaev I. Hardware and software means for fault-tolerance of computer system. *IBID* , No. 2, pp. 3-53, 1988.

[3] von Neumann J., Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: *Automata Studies, Ann. of Math. Studies No. 34* (C. E. Shannon and J. McCarthy, eds.), pp. 43-49. Princeton Univ. Press, Princeton, New Jersey

[4] Schagaev I., Yet another approach to classification of redundancy, Proc. FTSD, Prague, Czechoslovakia, May 1989, pp. 485-490.

[5] Schagaev I, J. Zalewski, Redundancy classification for Fault Tolerant Computer Design, *Proc. 2001 IEEE Systems, Man, and Cybernetics Conf.*, Tucson, AZ, October 7-10, 2001, Vol. 5, pp. 3193-3198.

[6] Avizienis A., The Star (Self-Testing and Repairing) computer: An Investigation of the Theory and Practice of Fault-Tolerant Computer Design, *IEEE Trans. on Computers*, Vol. C-20 (11), pp. 1312-1321.

[7] Avizienis A., Architectures of fault tolerant computing systems, *Proc. FTCS Symposium* , 1975, pp. 3-16.

[8] Avizienis A.,,.Ng Ying W, An Unified Reliability Model for Fault Tolerant Computers, *IEEE Trans. Computers* , Vol. C-29, No.11,.pp. 1002-1011, November 1980.

[9] Avizienis A., Laprie J., Dependable computing: from concepts to design diversity. *Proc. IEEE* , Vol. 74, No. 5, May 1986.

[10] Avizienis A., N-version approach to fault tolerance software. *IEEE Trans. on Software Eng ineering* , Vol. SE-11, No. 12, pp. 1491-1501, Dec. 1985.

[11] Laprie J.C., A. Costes, Dependability: a unifying computer for reliable computing, *Proc. 12th Int. Symp. on Fault tolerant computing systems, FTCS-12* , Los Angeles, June 1982, pp. 18-21.

[12] Laprie J.C., Dependable Computing and fault-tolerance: concepts and terminology, IFIP WG 104, Summer 1984 meeting, Kissimmee, Florida; *LAAS Research Report No. 84.035* , June 1984.

[13] Kopetz H. et al, Tolerating transient faults in MARS, *Proc. 20th Int'l. Symp. on Fault Tolerant Computing Systems* , Newcastle Upon Tyne, U.K., June 1990, pp. 466-473.

[14] Siewiorek D. Reliable *Computer Systems: Design and Evaluation.* Burlington, MA, Digital Press 1998.

[15] Stepanyants A., et al.. Malfunction Tolerant Processor and Its Reliability Analysis, *DSN 2001* , G ö teborg, Sweden.

[16] O'Brien F, Rollback point insertion strategies, *Proc. FCTS-6* , Pittsburgh, Penn., 138-142 (1978).

[17] Rendall B., System structure for software fault tolerance. *IEEE Trans. on Software Engineering* , .Vol. SE-1, No. 2, pp. 191-209, June 1975.

[18] Russell D.L, M.J. Tiedeman, Multiprocess recovery using conversations, Proc. FCTS-9, 1979, pp. 106-109.

[19] DeAngelis D,. Lauro J, Software recovery in the fault-tolerant spaceborne computer, Proc. FCTS-6, Pittsburgh, Penn., June 1976, p. 143.

[20] Wirth N., Gutknecht J.: *Project Oberon* . Addison-Wesley 1992.

[21] Schagaev I., Algorithms of Computation Recovery, . *Automatic and Remote Control*, 7, 1986.

[22] Schagaev I., Algorithms to Restoring a Computing Process, *Automatic and Remote Control,* 7, 1987.

[23] Schagaev I., Determination of type of hardware faults by software means. *Automatic and Remote Control,* 3, 1990.

[24] Vilenkin S., Schagaev I., Operating System for Fault Tolerant SIMD *Computers. Programmirovanie* , No.3, 1988 (In Russian).

[25] Pierce W.H., *Failure-Tolerant Computer Design* , Academic Press Inc New York, 1965.

[26] Birolini A., *Reliability Engineering Theory and Practice. 8e* , Springer, 2007.

# Rapid Control Prototyping with Scilab/Scicos/RTAI for PC-based and ARM-based Platforms

Skiba Grzegorz, Żabiński Tomasz, Bożek Andrzej
Rzeszów University of Technology
Email: skiba.g@gmail.com, tomz@prz-rzeszow.pl, bozekand@op.pl

*Abstract*—**This document describes three didactic systems with a Rapid Control Prototyping (RCP) suite based on Scilab/ Scicos and a PC computer. Servomechanisms with wire signal transmission and PID as well as a fuzzy controller are presented. The servomechanism with wireless signal transmission is also described. As the RCP suite based on Scilab/Scicos/RTAI was successfully used in the servo controller development on a PC platform, the concept of a tool-chain for RCP on an embedded platform with ARM processor is discussed. The TS-7300 embedded system with RTAI real-time operating system is described. The changes in Scilab/Scicos needed to interface the generated controller code to TS-7300 are presented. The didactic use and a possible commercial use of the tool-chain is indicated. The didactic use is focused on controllers tuning, friction and long delays influence on servomechanism control and embedded control systems development.**

## I. Introduction

Rapid Control Prototyping (RCP) gives a possibility for quick and convenient control strategy verification and iterative controller development. RCP involves a controller simulated in real-time, coupled with a real plant via hardware input/output devices. Nowadays RCP has become a crucial method in developing and testing control strategies within time acceptable by the market. It is also a very popular technology in scientific and didactic laboratories.

RCP requires two components: a Computer-Aided Control System Design (CACSD) software and a hardware with a hard real-time operating system. CACSD includes a broad range of computational tools and environments for: control systems design, real-time simulation, data acquisition and visualization. CACSD should support all the phases of the control system development namely specification, modeling, identification, controller design, simulation, implementation and verification [1], [2]. The most significant element of CACSD in supporting implementation is an integrated code generator which allows a direct creation of a controller code from e.g. a graphical schema. One of the most popular RCP systems is based on the Matlab/Simulink/RTW (Realtime-Workshop) suite [3] which enables creation and compilation of a controller code for different targets. The Matlab/Simulink/RTW commercial product is very popular at universities and in industries. The main advantage of the suite is its convenience and accurate graphical interface. The main drawback is given by its cost. An interesting alternative to Matlab/Simulink/RTW is the free and open-source Scilab/Scicos/RTAICodeGen suite [4]. The alternative is based on Scilab/Scicos, Linux RTAI and RTAI-Lab. The RCP open source environments are especially desirable for didactic purposes because of their low cost. The RCP tool-chain based on RTAI-Lab has been successfully used in Department of Computer and Control Engineering at the Rzeszów University of Technology. The tool-chain is used in Mechatronics, Control Theory, Embedded Systems and Real-Time Control courses. Additionally the tool-chain is also used by students and researchers who engage in the creation of control systems.

This paper describes the existing laboratory equipment for servo control with wire or wireless signal transmission. The examples of using PID and Fuzzy Model Reference Learning Controller (FMRLC) for a servo control are also briefly described in the paper.

The paper is organized as follows: in Section II the Scilab/ Scicos/RTAICodeGen suite is described; in Section III three different didactic control systems for servomechanisms are presented; in Section IV an idea of the RCP tool-chain for embedded system as well as the architecture of TS-7300 em bedded platform with ARM9 processor and FPGA chip is briefly described; and finally in Section V the main results are briefly discussed.

## II. Scilab/Scicos/ RTAICodeGen suite

Scilab is a scientific software for numerical computations. It has been developed since 1990 by scientists from INRIA (Institut National de Recherche on Informatique et on Au tomatique) [5] and ENPC (École Nationale des ponts et chaussée) [6]. Scilab provides a rich set of functions for sci entific applications and engineering [7].

Scicos is a Scilab toolbox which allows a graphical design of a control system. The design and simulation of a control schema can be realized directly from the Scicos graphical interface in a comparable manner to Simulink.

Linux RTAI is a hard real-time extension of the Linux Operating System. RTAI has been developed since 1999 by the researches from the DIAPM (Dipartimento di Ingegneria Aerospaziale del Politecnico di Milano) [8].

The RTAI environment has been extended to include a code generator called RTAICodeGen in order to obtain com patibility with RTAI for a hard real-time code from Scicos schemas.

RTAI-Lab is a tool included in an RTAI distribution which provides a framework for designing, building, running and monitoring RTAI-based controllers and real-time simulators. The controllers can be coded manually in languages akin to C/C++ or generated automatically by Scilab/ Scicos/ RTAICodeGen or by Matlab/Simulink/RTW. The RTAI-Lab framework contains block library that allows the interaction of Scicos schem a s with a data acquisition hardware. Drivers for acquisition hardware are provided by the COMEDI project [9] or can be implemented by the user [4]. *Xrtailab* the GUI application included in RTAI-Lab is used for the purpose of data acquisition and monitoring and so as to change controller parameters on the fly.

The generated controller typically runs as a user space hard real-time application on a standard PC computer [1], [10]. Nowadays it is a typical way of using the RCP suit based on Scilab/Scicos and RTAI. As RTAI supports the following hardware architectures:

- x86 and x86_64
- PowerPC
- ARM (some subarchitectures)

it would be convenient to use the RCP suite for Rapid Control Prototyping on a destination embedded hardware [11].

Section IV briefly describes the way to obtain the RCP tool-chain for the TS-7300 embedded system with the AR M920T processor.

## III. Control of a DC servo motor in Scilab/Scicos/RTAI-Lab

Three didactic servo control systems developed in Department of Computer and Control Engineering are described in this section. The systems are used for didactic purposes in courses of Mechatronics and Control Theory.

### *A. Plant description*

The plant includes a DC servo motor from Pittman, an HP incremental encoder, an MSA-12-80 power amplifier in current mode from Galil and a KR33 linear stage with a ball screw from THK (Fig. 1).

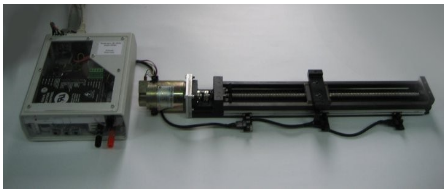


Fig. 1. Laboratory servomechanism

For RCP purposes a general PC computer equipped with Linux RTAI as real-time operating system is used. For wire signal transmission the input/output board RT-DAC4-PCI [12] from Inteco is used. The driver for RT-DAC4-PCI card was developed using COMEDI specification [4]. In wireless communication two transmission modules are used: one connected to the power module and the second connected to the PC computer via the RS232 interface. The transmission modules were constructed using ADuC7128 controllers and TLX2401 radio modules [10].

### *B. DC servo control with wire signal transmission and PID controller*

In order to use the RT-DAC4-PCI card in real-time experiments the driver and communication blocks for Scicos were developed [4]. The blocks for: A/D and D/A conversion, PWM outputs and encoder counters were developed (Fig. 2).

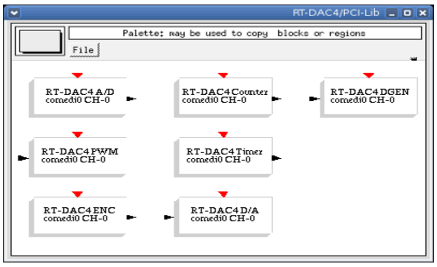

RT-DAC4/PCI-Lib
Palette: may be used to copy blocks or regions
File
RT-DAC4 A/D comedi0 CH-0
RT-DAC4 Counter comedi0 CH-0
RT-DAC4 DGEN comedi0 CH-0
RT-DAC4 PWM comedi0 CH-0
RT-DAC4 Timer comedi0 CH-0
RT-DAC4 ENC comedi0 CH-0
RT-DAC4 D/A comedi0 CH-0

Fig. 2. Communication blocks for RT-DAC4-PCI and Scicos

Control system structure with PID digital controller is presented on Fig. 3.

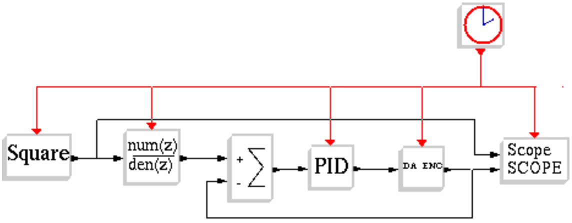


Fig. 3. Control system structure for DC servo control with PID discrete controller

The block DA-ENC includes the D/A conversion block and the encoder counter interface block which provides communication with the real plant (Fig. 4).

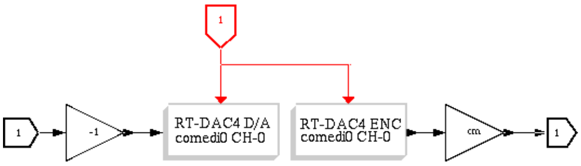


Fig. 4. DA-ENC block internal structure

The PID block internal structure is presented on Fig. 5.

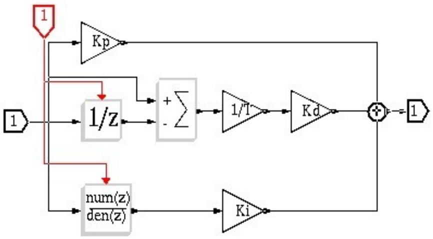
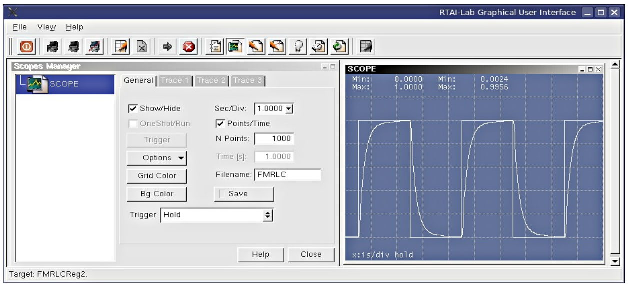


Fig. 5. Discrete PID controller structure

The system is mainly used for didactic purposes. The students perform identification procedure of the real-plant model and perform PD and PID controllers tuning. Many different experiments are performed in order to test the step response parameters and the influence of friction on servomechanism systems e.g. steady-state errors, hunting and stick-slip.

A plant model given by a double integrator transfer function is identified via a step response real-time experiment conducted using Scicos schema. The Scilab *datafit* function is used to fit the model to the experimental data. The PD and PID controller is tuned using root-locus design method. In this case, two design data, i.e. gain of the drive treated as double integrator and required settling time, are only needed [13]. The results of servomechanism real-time step response experiment are presented on Fig. 6.



☑ Show/Hide
□ OneShot/Run
Sec/Div: 1.0000
☑ Points/Time
N Points: 1000
Time [s]: 1.0000
Filename: FMRLC
Trigger: Hold

```
Min:     0.0000    Min:     0.0024
Max:     1.0000    Max:     0.9956
x:1s/div hold
```

Target: FMRLCReg2.

Fig. 6. Servomechanism real-time step response experiment with PID controller - *xrtailab*

The obtained results are similar to the ones which use the RCP suite based on Matlab/Simulink/RTW and RT-CON [12].

### C. DC servo control with wire signal transmission and FMRLC controller

A more sophisticated controller for a servomechanism with Fuzzy Model Reference Learning Controller (FMRLC) [14] is also used by the students [4]. The control system consists of the reference model, fuzzy Takagi-Sugeno controller under learning process, fuzzy inverse model, learning mechanism and knowledge-base (Fig. 7). The two inputs to the fuzzy PD controller are the error *e* and change in error *de* (Fig. 7). On the basis of the internal knowledge the controller produces control signal *u* which is fed to the plant (power amplifier). The fuzzy controller is continuously taught during the control process by a fuzzy inverse model, learning mechanism and knowledge base. The learning goal is to keep a plant output ($y$) as close as possible to the reference model response ($y_m$). Only the consequents of the controller rules are changed, fuzzy sets remain unchanged.

The Scicos control schema with FMRLC controller is shown on Fig. 8.

The FMRLC block was developed in C language as a computational function for Scicos [4], [7].

The system is used for didactic and research purposes. The students perform experiments with different configurations of FMRLC e.g. simple fuzzy controller in control loop and complete FMRLC structure.

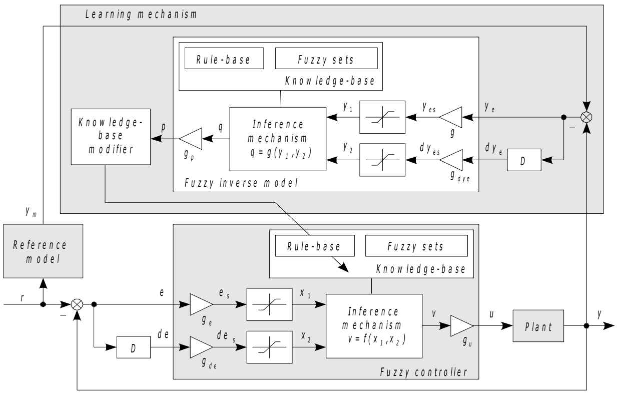


Fig. 7. FMRLC structure [14]

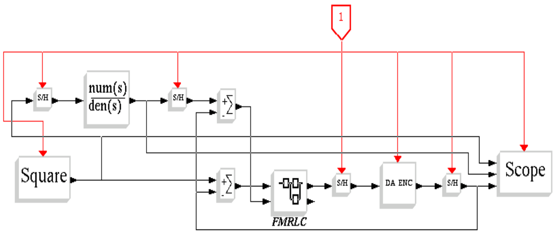


Fig. 8. Scicos schema with FMRLC

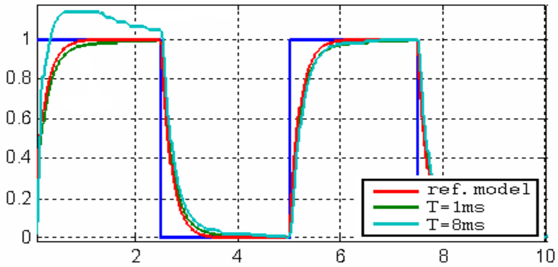


Fig. 9. Servomechanism real-time step response experiment with full FMRLC configuration

On Fig. 9 demonstrative results of the experiments are presented.

For different control periods $T$ the servomechanism gives different results (Fig. 9). As it is shown on Fig. 9 the system response with $T$=1 ms is very close to the model reference response. It must be emphasized that FMRLC rapidly gains the knowledge of how to control the plant. At the beginning of the experiments all the rules consequents were set to zero.

### D. DC servo control with wireless signal transmission

The third example of using Scilab/Scicos/RTAICodeGen, RTAI and RTAI-Lab for servo control is described in this section. Unlike the already described systems this one uses wireless transmission (Fig. 10) between a controller and an object.

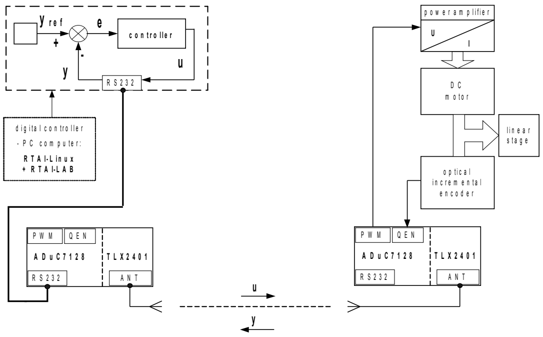


Fig. 10 . Structure of servomechanism with wireless signal transmis sion

In the system, instead of the RT-DAC4/PCI card, transmission via the RS232 interface and two modules with the ADuC7128 controllers and the TLX2401 radio modules are used. The modules make bidirectional wireless transmission possible for PWM control signal, position measurements and additional information. The structure of the communication frames can be freely arranged by the module software. Special blocks were created for Scicos, designed for a real-time communication via the RS323 interface and signal multiplexing/demultiplexing. These blocks allow the integration of the developed hardware with the RCP software environment. The real plant with transmission modules is presented on Fig. 11.

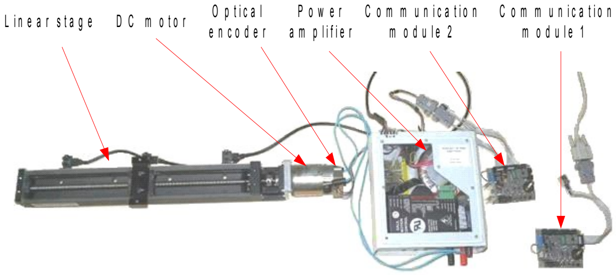


Fig. 11. Servomechanism with wireless signal transmission

The system was primarily designed as a didactic equipment used to test control algorithms in case of long delays or data loss. A few simple solutions are tested by students:

1. Keeping the last control value active until the next one arrives (zero order hold) versus resetting (setting to zero) control while data does not arrive.
2. Extension control period over many communication periods.
3. Using predictive control calculation based on object model equation on the controller side.

The example of Scicos schema for the control system with the PID discrete controller and anti-wind-up system is shown on Fig. 12. The control period for control loop is 3 ms and there is a possibility to count the number of lost data packets.

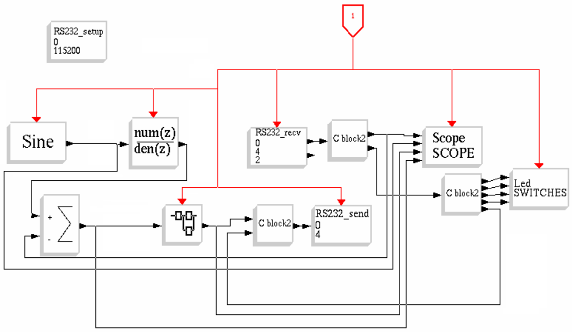


Fig. 12. Scicos schema for wireless servomechanism with PID controller

On Fig. 13 the demonstrative results of the experiments are shown. The reference model response reflects the desired response shape obtained from a simulation model without wireless communication. Remote plant responses were registered in the case of approximately 25% and 75% loss of data packets during the experiment. The level of the data loss was tuned by modifying the distance between the modules and placing barriers on waves propagation way. The degradation of responses quality according to higher number of lost packets is observed.

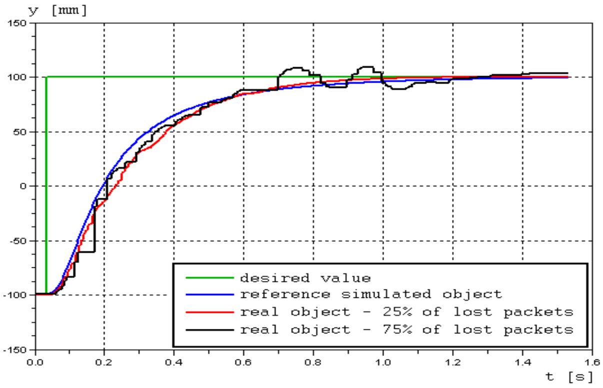


Fig. 13. Wireless servomechanism real-time step response experiment with PID controller

The remote communication module can also simulate object transfer function. There are a few advantages of this solution for students, e.g.:

1. It is possible to control objects that are not physically available in the laboratory.
2. Object model can be precisely defined and is not disrupted by unmodeled real plant dynamics. It allows students to concentrate on long delays or data loss compensation.

The transfer function given for a simulated object can be expressed in *s* or *z* domain. If the transfer function is given in *s* domain it is converted to *z* domain in the Scicos context script (automated action) applying a bilinear transformation. Finally, coefficients of the transfer function are inserted into a specially designed Scicos block ( `ControlSetup` ) that sends them to the remote communication module when the controller application starts. The process was automated as much as possible using the RCP environment. Only information that must be directly given by a student (object transfer function and controller diagram) is inputted manually.

Fig. 14 presents the demonstrative results of the system step response with a remote simulated object given by the transfer function $G(s)=\frac{36}{s^2-36}$ .

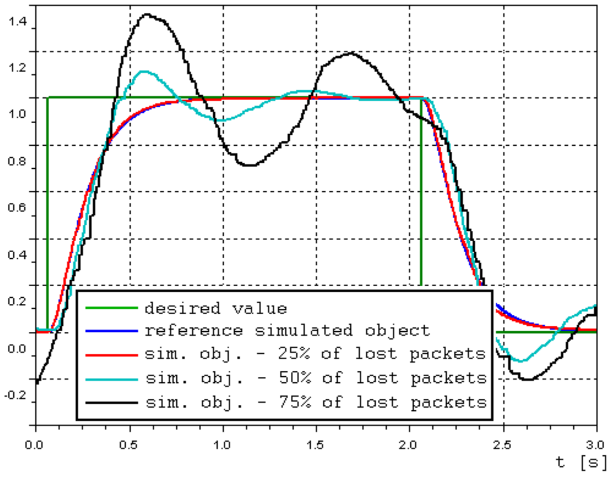


Fig. 14. The system step response with the remote simulated object

The three examples described in this section show the possibility of using Scilab/Scicos and RTAI for educational purposes. The mechatronics group in the Division of Informatics and Control at the Rzeszów University of Technology has the experience in using the Matlab/Simulink/RTW suit [15]–[17] as well as Scilab/Scicos/RTAI [4], [10]. Positive experience in using Scilab/Scicos/RTAI generated the definition of a new research goal. The goal is to build tool-chain for RCP on destination embedded hardware TS-7300 with the ARM9 processor and the FPGA chip.

## IV. RCP TOOL-CHAIN IDEA FOR EMBEDDED SYSTEM

### A. *Tool-chain idea*

Typical use of the Scilab/Scicos/RTAI suit involves a PC computer as a hardware platform in order to develop, run and test controllers. The goal defined in the mechatronics group in the Division of Informatics and Control assumes developing RCP tool-chain for embedded platform TS-7300. PC computer will be used as a platform for: developing controllers in Scilab/Scicos, cross-compiling controller code for embedded target, data acquisition and visualization. The executable controller code will be automatically deployed and run on the embedded platform. The GUI interface will be used on a PC computer for visualization and data acquisition. The sequence of actions taken up when tool-chain will be used is presented on Fig. 15.

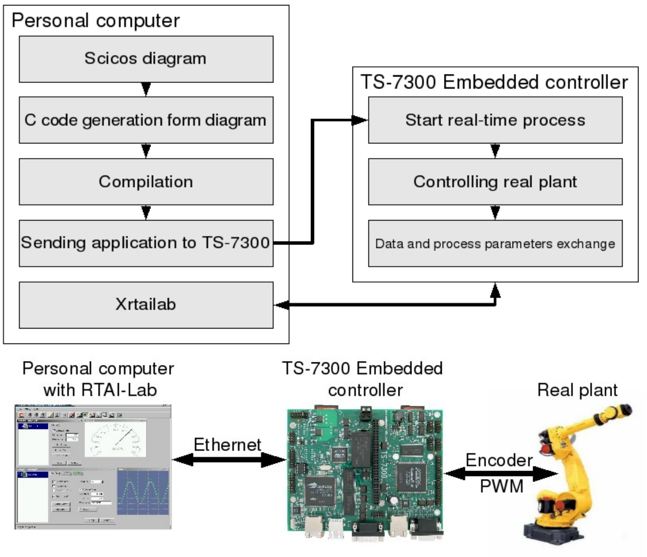

Personal computer
Scicos diagram
C code generation form diagram
Compilation
Sending application to TS-7300
Xrtailab
TS-7300 Embedded controller
Start real-time process
Controlling real plant
Data and process parameters exchange
Personal computer with RTAI-Lab
TS-7300 Embedded controller
Real plant
Ethernet
Encoder
PWM

Fig. 15. Tool-chain for embedded system

In order to obtain remote GUI interface, middle-ware layer based on UDP [2] is going to be used.

To achieve the goal the following main problems are de fined to be solve:

1. Recompilation of the RTAI kernel in order to enable floating point operations in real-time tasks. TS-7300 is equipped with MaverickCrunch math co-processor.
2. Modification of automatic code generator in order to support network communication between *xrtailab* and real-time tasks without *net-rpc* mechanism which is not available for the ARM platform.
3. Developing Scicos blocks for input/output operations using encoder counters and PWM outputs implemented in the FPGA chip.

Embedded platforms are considered to be more proper for a real-time control than general PC machines. Therefore, the RCP for embedded system can be used for didactic and in dustrial purposes.

The created tool-chain will be used for developing control systems for CNC machines and laboratory robots. The CNC machine control system based on RT-Linux [18] will be exchanged with newly developed system using the tool-chain. The tool-chain will be also used for didactic purposes i n courses of Embedded Systems and Real-Time Control.

### B. *The TS-7300 embedded platform*

The embedded platform consists of the TS-7300 Single Board Computer (Fig. 16) equipped with the RTAI real-time operating system.

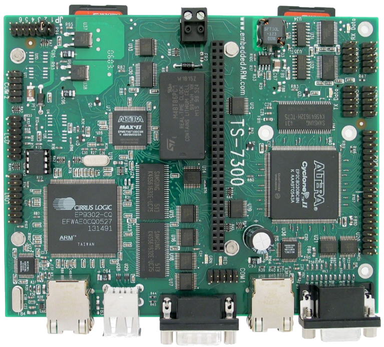


Fig. 16. TS-7300 Single Board Computer

The TS-7300 embedded system is a multipurpose board with a Cirrus Logic EP9302 200MHz processor. EP9302 features an advanced ARM920T processor designed with a memory management unit (MMU) that allows the support of high level operating systems such as Linux with a real-time extension. The modern processor connected via Wishbone, an open source hardware high-speed bus with Altera FPGA increases the flexibility of the system. The FPGA allows the development of user defined logic which can perform common controller tasks that are difficult, CPU intensive, or impossible to accomplish in software with regular DIO/GPIO hardware and the facilities of the processor. The TS-7300 provides up to 35 DIO lines connected directly to the FPGA through DIO header which can be used as a interface for control applications. In order to create servo controller, a student has to change the default FPGA configuration by implementing encoder counters and PWM generators. Peripherals such as SD Card flash socket, VGA controller, 2 USB ports, 10/100 Ethernet ports, RS232 serial ports are also included on the board.

In order to operate on the board it is possible to connect a mouse, keyboard and a monitor to USB and VGA connectors. The board can be used with additional laptop and SSH via Ethernet. For files transfer SCP software can be employed.

In order to adapt TS-7300 for servo control purposes as well as for cooperation with the Scilab/Scicos/RTAI suite the following problems were solved:

1. A cross-compiler for the embedded platform was developed. The created cross-compiler supports C, C++ and Fortran programming languages. The support for Fortran is required for automatic code generation from Scicos schemas. The cross-compiler delivered with TS-7300 does not support Fortran.
2. The RTAI kernel delivered with TS-7300 was modified in order to allow running real-time tasks in a user mode. In the original version of kernel delivered with TS-7300 a real-time task running in user mode caused system crash.
3. Encoder counters and PWM outputs have been implemented using Verilog language and Quartus II software tool in FPGA chip.

The tool-chain will let students to graphically design control structure, simulate system with plant model, deploy controller code into embedded system, run and test system for real-plant control.

## V. Conclusion

Scilab/Scicos/RTAI is low cost, high quality and an open source Rapid Prototyping System. The system can be used on courses of Control Theory, Embedded Systems, Real-Time Control and Mechatronics as an alternative to Matlab/Simulink. The system can be also applied in control system development for embedded platforms with ARM processors. It seems that the number of embedded industrial applications developed using Scilab/Scicos/RTAI should rapidly grow in the nearest future. According to the above mentioned expec tation teaching students applying Scilab/Scicos/RTAI in the development of embedded control systems is indispensable.

## References

[1] R. Bucher, L. Dozio, *CACSD with Linux RTAI and RTAI-Lab*, Valencia [Online], 2003, ftp://ftp.realtimelinuxfoundation.org/pub/events/rtlws-2003/proc/bucher.pdf

[2] R. Bucher, L. Dozio, *Rapid Control Prototyping with Scialb/Scicos and Linux RTAI* , http://www.scilab.org/events/scilab2004/final_paper/1 - bucher_rtai2scilab.pdf

[3] http://www.mathworks.com

[4] G. Skiba, T. Żabiński, K. Wiktorowicz, *Rapid prototyping of servo controllers in RTAI-Lab* , VI Conference on Computer Methods and Systems, 141–146, Cracow 21-23 November 2007.

[5] http://www.inria.fr

[6] http://www.enpc.fr

[7] S. L. Campbell, J. P. Chancelier, R. Nikoukhah: *Modelling and Simulation in Scilab/Scicos* , Springer 2006.

[8] http://www.rtai.org

[9] http://www.comedi.org

[10] A. Bożek, G. Skiba, T. Żabiński, *Rapid prototyping of control systems in RTAI-Lab* , X Krajowa Konferencja Robotyki (to be published), Piechowice 3–6 September 2008.

[11] M. Petko, T. Uhl, *Embedded controller design – mechatronic approach* , Proc. Of the Second International Workshop on Robot Motion and Control RoMoCo 2001, pp. 195–200.

[12] http://www.inteco.cc.pl

[13] T. Żabiński, *Tuning PID controllers for servo* , PAR, vol 4 (134), 56-63, 2008.

[14] K. M. Passino, S. Yurkowicz, *Fuzzy Control* , Addison-Wesley, 1998.

[15] D. Marchewka, T. Żabiński, *Adaptive neural controller for the task of trajectory tracking of the 3DOF manipulator* , VI Conference on Computer Methods and Systems, 195–200, Cracow 21-23 Novem ber 2007.

[16] T. Żabiński, T. Grygiel, B. Kwolek, *Design and implementation of visual feedback for an active tracking* , ICCVG, Warsaw, Machine Graphics & Vision, 2006.

[17] T. Żabiński, A. Turnau, *Compensation of friction in robotic arms and slide tables* , 16 th IFAC World Congress, Prague, July 4 to July 8 2005.

[18] T. Żabiński, *Control of mechatronics systems in real-time – classical and intelligent approach* , PhD thesis, AGH University of Science and Technology, Cracow 2006.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 745 – 750

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Development of a Flight Control System for an Ultralight Airplane

Vilem Srovnal Jr, Jiri Kotzian
VSB – Technical University of Ostrava
Department of Measurement and Control, FEECS
17. listopadu 15, 708 33, Ostrava – Poruba,
Czech Republic
Email: {vilem.srovnal1, jiri.kotzian}@vsb.cz

*Abstract*—**This paper presents development of the hardware and software for the low cost avionic system of ultralight airplanes. There are shown three levels of a hardware and software system design. As far as the software is concerned, we focused on changeover from non real-time operating system (embedded Linux) to the real-time embedded platform (QNX). We discussed the problems that led to operating system change. Various advantages and disadvantages of both operating systems are presented in this contribution. Concerning the hardware we concentrated on the development of the avionic control, monitoring and display modules that are a components of the dash board. The paper has been focusing on safety and reliability in the ultralight aviation and what can be improved or extended by the real-time operating system.**

## I. Introduction

THIS paper shows dependence hardware and software reliability. The control and monitoring system for ultralight or sport airplane has to be stable in an extreme situation as well as in a daily routine. When one designs your own hardware and software architecture one has to keep in mind the hazard states that can occur in many cases. In the aviation industry the safety and reliability are very important goals. But the other aspect is a product price. It means that hardware for double or triple protection is too expensive for the customer so we have to presume that hardware will work reliably at all times. This is the same case of our system that is why we focus on software architecture development in this article. The embedded Linux as main operating system has been implemented in our embedded system. The basic problem are drivers that have to be certificated and be very safe as all operating system. On the Linux platform it is difficult to ensure that drivers do not corrupt the kernel because both run within the same address space as the OS kernel. Therefore we were looking for a new software architecture solution. As the ideal answer it resulted in QNX real-time platform with client-server architecture where drivers are independent on the kernel (microkernel) that only implements the core services, like threads, signals, message passing, synchronization, scheduling and timer services. There are other possible real-time operating system of course that are very often use in the avionic systems, for example VxWorks from Wind River or Integrity from Green Hills but the problem for the low cost product is license price of the RTOS. When the run-time license price of RTOS is over the client limits there is no way how to use them. Especially Integrity software is very powerful system for developing purpose on the different platforms and has a very nice debugging tools.

## II. RTOS Selection

The right choice of a software system architecture is important for following development of all stuff. Due to the lack of our experiences and that requirements on the system where growing up during developing period, it caused that we had to try three different architectures.

### *A. Embedded uClinux*

The first platform was the embedded uClinux based on microprocessor ColdFire MCF5329 from Freescale. This solution had shown us that graphic hardware is not powerful enough as we supposed and the lack of MMU was the huge problem as well. [3] On the other hand the advantage of this solution is quick access to memory and integrated CAN hardware device on the developing board. The address space

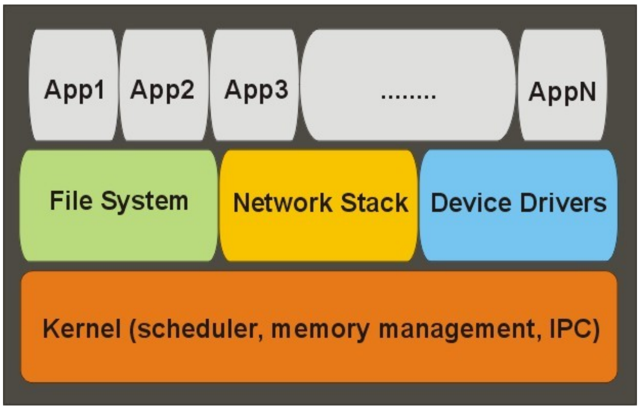


Fig 1. Architecture of uClinux - traditional for old RTOS

This work is supported by grant of the Grant Agency of the Czech Republic GA102/08/1429 - Safety and security of networked embedded system applications

is common for the kernel  as well as for the user application which decreased safety and reliability of the entire control system but it is real-time in nature because there is no overhead of system calls, message passing, or copying of data. The uClinux architecture (flat addressing model) is shown on the  Figure 1.

### B. *Embedded Linux*

The embedded Linux architecture was chosen as the next platform for avionic system. This architecture is based on monolithic kernel that has distinction between the user and kernel space. [5] Any fault in the application will not cause system crash. The system has run on ARM processor PXA 270 from Intel and ARM processor PXA 320 from Marvell. The  both processors are very powerful but there is a lack of floating point instructions set that is very important for graphic operations. The next problem that was mentioned above is incorrectly written driver or module that can cause the system to crash. [6] The embedded Linux architecture is shown on Figure 2.

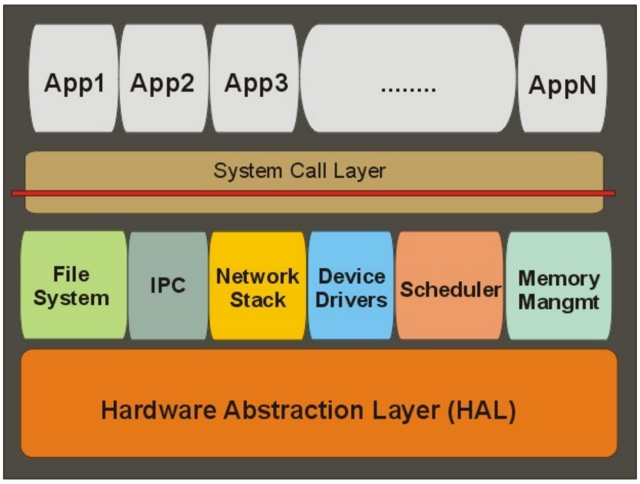


Fig 2. Architecture of embedded Linux

### C. *QNX Neutrino real-time OS V 6.3*

The embedded Linux has worked without problems but there was demand on system to be high reliable and safe. The embedded Linux was sufficient for the monitoring of avionic values but for the control activity it was worse. We had to ensure time death line for the control process activity and graphic smooth step to the next frame on LCD display. There were requirements to isolate graphic process from control process but in monolithic kernel architecture the graphic driver could corrupt all system. [4] So the next logical step was to use microkernel to realize mentioned requirement. The basic principle of microkernel architecture is that each of kernel subsystem (network stack, file system, scheduler, etc.) has private address space similar to application. This way offers complete memory protection, not only for user applications, but also for OS components. This architecture provides maximum modularity and relies on robust message passing schema. The QNX microkernel architecture is shown on Figure 3. [11]

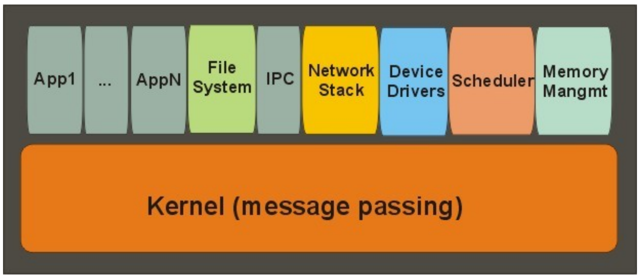


Fig 3. Architecture of QNX microkernel

Due to the lack of floating point instructions set we needed to use another processor that supports FPU and QNX RTOS. We decided to use ARM microprocessor iMX31 from Freescale but parallel solution stop on ARM processor PXA 270 with QNX RTOS. Unfortunately for the ARM microprocessor PXA 320 was not QNX BSP available.

## III. Software Architecture

The reliability is main goal in the process control of the most industrial or others systems. As we presented in software architecture part, there are many reasons, why divide all control and monitoring system into the smaller software distributed units – processes.

The client-server architecture brings features that  are able to guarantee more stability of the whole system. The failure of one part of the system (process) causes crash of any other one. The interprocess communication is based on robust message passing. The processes are separate according its functionality and hardware requirement. It means that each process uses one specific driver. Every individual process runs in independent address space.

There are five basic process groups. The first one is the motor control group that is responsible for everything what is connected with plane control. We measure avionic sensors inputs that are needed for autopilot control, maximum and minimum plane speed, etc.

The second group takes care of communication with other hardware parts. We are using CAN communication protocol and CAN driver where each hardware device is CAN node. [2] There is possible to use serial communication RS232 or USB as special process as well but for another purpose.

The third big group is SCADA system. The SCADA system or we can say graphic tasks that are responsible for monitoring and display avionic values in continuous time (it depend on processor time scheduling). Graphic process is very demanding on time of the processor so the priority level are low but have to ensure specific frame count per second.

The fourth group is IO device control for other purpose. We use GPIO driver for pins states monitoring. Each pin has special function where the most of them are used as button, trimer or switch. We use GPIO-event handling mechanism for pin state tracking.

The fifth group and the last process group is audio device group that is responsible for voice transmitting into the central communication system. There is audio system for warning and critical errors as well. The audio system can be used for other purpose it is up to client additional extension. The

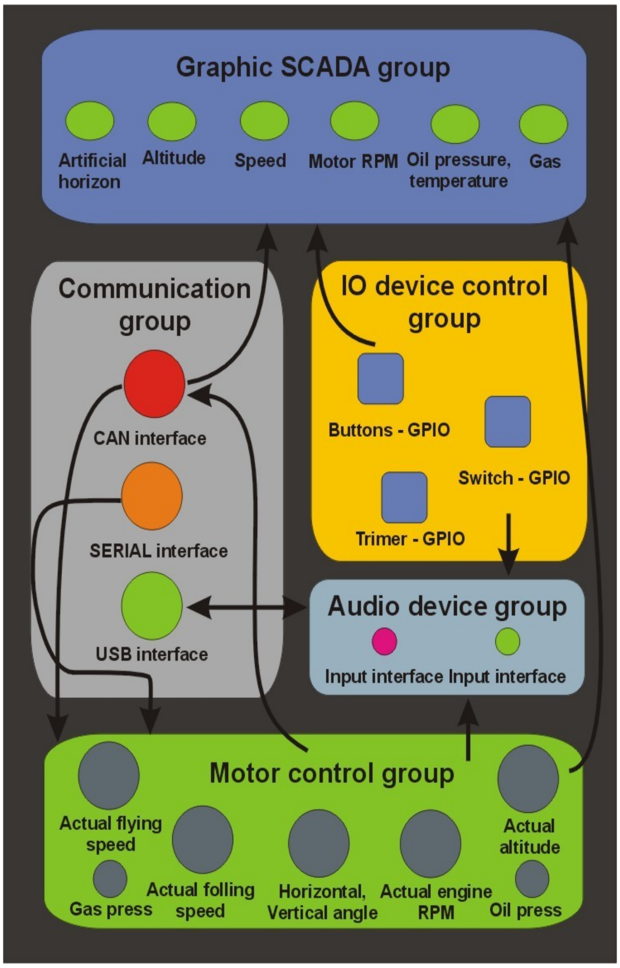


Fig 4. Architecture of process control

following schema on the Figure 4 shows process structure. The communication group allows data for other process groups from sensors and actuators. The motor control group uses communication group for engine values control. The communication unit has to be one of the most reliable part of the control and monitoring system.

## IV. Critical System Control

The basic critical processes of the aviation control have to meet the following criteria. The plane take-off speed as well as landing speed has to meet the required characteristics. These values are measured with sensors which are build-in the hardware board. The measurements are completed by the pressure difference against the surrounding environment. All data which have been gained from the sensors are used for the rotation engine process. The measurement of the stalling speed is a very important to prevent the uncontrolled plane fall /accident/ and jet-engine failure which is closely connected with this event. The measurement of the vertical and horizontal plane heeling is the next critical process. Exceeding the allowed plane heeling can result in stability loss of the airplane. To make the jet-engine work sufficiently the pressure in the gas and oil tank must be sustained. Other important flying data are information about a plane position. The position is shown through GPS system. All above mentioned processes must run in the independent address space so that the safety of the system is sustained.

The driver design is very important for communication and stable control of the application. The critical data goes through CAN interface which gives or receives information for engine control and fly control. The CAN driver is connection between module control node (present in next chapter) and motor control process group. The next sophisticated driver which is used for SCADA system is framebuffer driver. This driver has to be very efficient and quick. The visualization system was extended by OpenGL ES 2D driver [12]

## V. Hardware Architecture for Avionic Systems

The avionic system is distributed into the independent modules that measure specific values on mechanical parts of the airplane. The hardware solution is a configurable according type of airplane. The highest layer is graphic user module that represents received data on the LCD display. The sense of monitoring system is offer customers same facilities as have pilots in the professional aircrafts and make aviation more easier using low cost embedded electronic system.

The real-time embedded control system is designed with a modular structure. [4] This structure supports a flexible configuration. In terms of user requirements, the control system can be configured in different sizes and options. [8] The s everal modules with different options were designed. All modules are connected to an industrial bus – so each module is the bus node. Except the GPS module that is connected directly to the main control module.

This architecture supports a future expansion. In terms of user requirements it is possible to design new modules. The new module node will be connected to the bus. The new module can work to satisfy a user after upgrading of the firmware in the main control mode. This way it is possible to connect a maximal 30 modules - nodes. The block diagram of a desk control and monitoring system with today's full configuration of prototype is shown on the Figure 5.

The monitoring and control modules are connected together by using an industrial bus [2]. This bus has to be highly reliable and have enough speed. Depending on these two main requirements a CAN bus was selected. The main reason is that the CAN has an extremely low probability of non-detected error. The versatility of the CAN system has proven itself useful in other applications, including industrial automation as well, anywhere that a network is needed to allow controllers to communicate. A CAN bus is given the international standard ISO11898 which uses the first two layers of ISO/OSI model (CAN-CIA 2005). The CAN is a multi-master protocol. When a CAN message is transmitted over the network all nodes connected to the network will receive the message. [7] Any node can begin transmitting at any time the bus is free of traffic and all nodes will listen to the message. Each node may employ a filtering scheme that allows it to process only relevant information. Each message has either an 11 bit identifier or 29 bit identifier which will define which node will receive the message, error checking bits, 8 bytes of data and priority information. If two nodes try transmitting a message at the same time, the node with the higher identifier (lower priority) loses control of the bus as soon as another node exerts a dominant bit on the bus. It ensures that the message with the highest priority will be transmitted on the first try.

**Designed modules are the following:**

- Main control module
- User interface (LCD display) module
- Motor measuring values module
- Advanced avionic data module
- Black-Box module

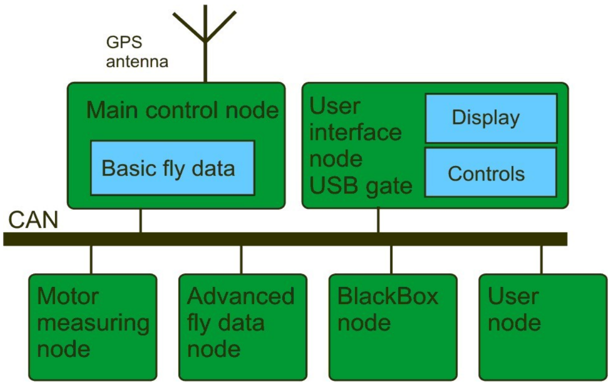


Fig 5. Block diagram of the control and monitoring system

### A. Main control module

The main control node serves as master for all other nodes. (Figure 6) [1] The requesting values from other nodes are compared with given limits and stored in the local memory. The main module decides what will be displayed and sends information to the user interface module by the CAN bus. The main control module contains a real time clock and data flash memory for storing measured values and statistics. For a measuring of the basic values the main module is equipped by sensors.

### B. Engine monitoring and control module

The engine monitoring and control module supports the complete monitoring and control of all necessary engine data. It collects all possible temperatures, pressures and the diagnostics of the engine. The basic motion control values and critical avionic data are already connected to the main control module.

### C. Avionic monitoring module

The advanced avionic data are supported by this module, for example a magnetic compass. This module only increase basic monitoring and control data that are supported in the main control module. It is up to client requirement.

### D. Black-Box module

The black-box module controls all traffic on the CAN bus. It reads data from CAN messages and stores data in the local memory. The black-box module is equipped with its own RTC timer and stores time together with the CAN data. There is no other connection to this module with such high reliability. [10]

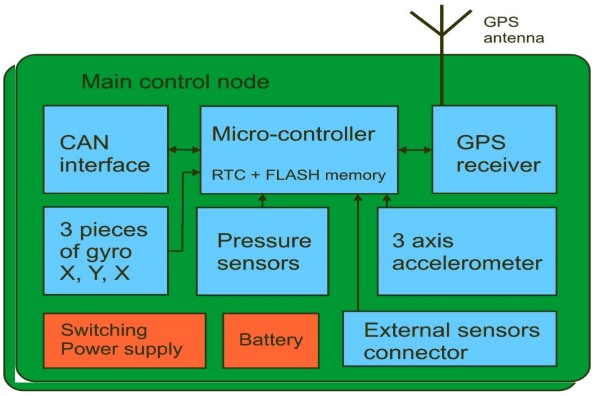


Fig 6. Main control node block diagram

### E. Development kits

There were three development kits used during the whole evolution. The first development kit that has been used comes from the world-wide known company – Freescale. This development kit is based on 32.bit microprocessors ColdFire family. This microprocessor runs on frequency 240 MHz and integrates display controller and other useful interfaces. [9] The second development kit that has been chosen comes from the Toradex company and integrates ARM microprocessor Intel PXA270 (Marvell PXA320) runs on frequency 520MHz respective for PXA320 on frequency 806MHz. There is an integrated display driver and CAN driver as well as other interfaces. [10] The third development kit that has been tested comes from company Logic PD. There is Freescale ARM microprocessor i.MX31 as SOM module integrated. This microprocessor runs on frequency 532MHz and integrates display controller and floating point unit as well.

### F. Prototype hardware development solution

The prototyping is the long process of improvements and cost lot of money and time. We have focused on finishing development on the ARM processors (Intel, Marvell and Freescale). The prototype board is designed for SOM module with processor PXA270 and PXA320. We are planning to design a new prototype board for SOM module based on processor iMX31. There is possibility to use other SOM modules for our prototype board as PXA300 or PXA255 as well.

These SOM module are intended for graphic processing of avionic data. The measurement and control module is based on 16.bit processor HCS12 from Freescale. There is not any RTOS integrated within. A lot of main control functions have been moved to SOM module (PXA270, iMX31) with RTOS QNX. The prototype implements interfaces as CAN, RS232, USB, GPIO. The standard interfaces like SERIAL and USB are supported by QNX BSP package but CAN and GPIO drivers have been developed by our software team to ensure drivers stability and reliability. We have chosen standard CAN controller SJA1000 produced by the Philips company. The GPIO driver has been implemented using older

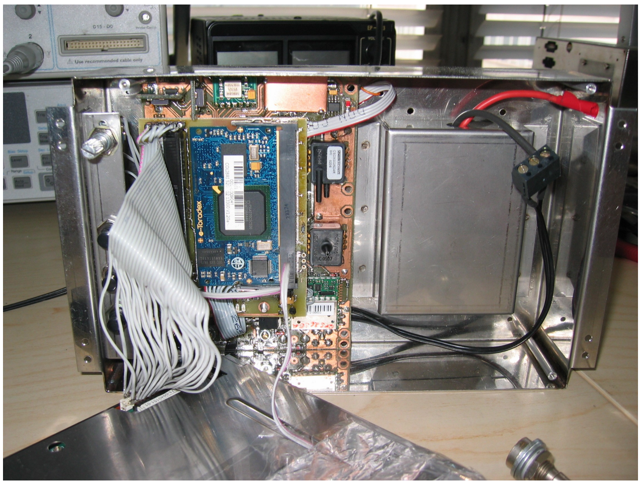


Fig 7. Prototype hardware realization

version where only address mapping had to be modified. The avionic system has three level of power supply. The first is standard supply from airplane battery that is recharged during a flight. When, due to some reasons or troubles this source is out of order, there are two batteries backup system. At the moment only one is embedded inside the box. The power supply for this prototype solution is 12V DC. The same power supply is used for LCD TFT display as well. The LCD TFT display has external backlight unit for better graphics recognition. The most energy from the power conception is consumed by the backlight module and LCD TFT display nearly more then 80% by the whole system. An extra extension can be reconfigured by the jumpers on the prototype board. Following Figure 7 and Figure 8 shows prototype realization.

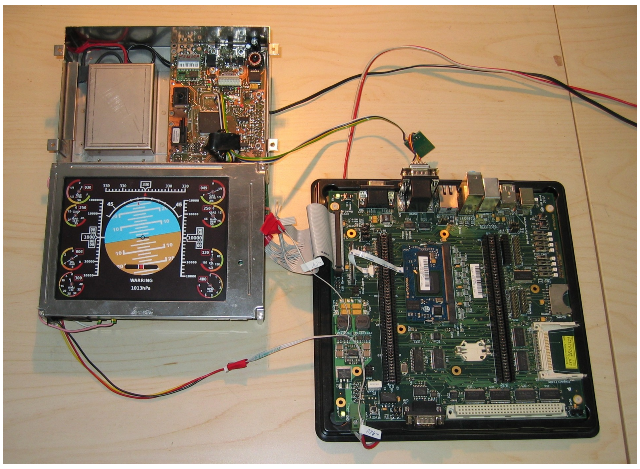


Fig 8. Prototype hardware realization with development kit

## VI. Graphic System Design

The graphic system design for the embedded system is usually quite difficult to meet optimal parameters for the mobile devices. The prototype solution that we proposed is embedded in the global avionic system and runs continuously during the flight. There is a big problem how to ensure low power consumption when you have to supply backlight module and LCD display. We can control backlight using software settings but this is not a cellphone where it is possible to switch off display when not used. But there are special cases when the backlight is switch off. For example when main power supply has to be replaced by battery and system controls backlight directly.

The PXA processors 270 and 320 lack of floating point unit and there is only floating point emulation that is very slow. The graphic operations base on emulation floating point are five times slower then on a fixed math primer. The fixed mathematics operations have been based on 16.bits integer scale what is sufficient for a precision of our application. The graphics rendering process takes a lot of processor time. So we decided to integrate graphic acceleration chip 2700G5 by the Intel company. The graphic acceleration has increased  the whole graphic process more then 100 times. The new processor PXA320 has integrated 2D graphics accelerator inside but the stable driver is not available yet. On the other hand the processor iMX31 by the Freescale company support 3D acceleration using their integrated chip inside and the driver is available for Linux and for QNX as well . The graphic library that has been used is OpenGL for embedded systems. There were only 2D graphic operations use in our system application. This OpenGL commands have to meet the avionic standards according OpenGL SC (Safety Critical for Avionics). The special example of 2D object rotation based on OpenGL is the gyro-horizon.[12]

## VI. Conclusion

The main goal of this paper is to show development and realization of the low cost avionic control and monitoring system for ultralight planes. We discussed safety and reliability of the control processes. In the step by step procedure we described software development process as well as hardware development process. The whole system was presented as distributed control system that consists of a few modules interconnected using CAN bus interface. We presented makeover from common operating system Linux to real-time operating system QNX.

## References

[1] Arnold K.: *Embedded Controller Hardware Design.* LLH Technology Publishing, USA, 2001, ISBN 1-878707-52-3.

[2] Sridhar T.: *Design Embedded Communications Software* . CMP Books, San Francisco, USA, 2003, ISBN 1-57820-125-X.

[3] Raghavan P., Lad A., Neelakandan S.: *Embedded Linux System Design and Development* . Auerbach Publication, New York, USA, 2006, ISBN 0-8493-4058-6.

[4] Li Q., Yao C.: *Real-Time Concepts for Embedded Systems* , CMP Books, San Francisco, USA, 2003, ISBN 1-57820-124-1.

[5] Hollabaugh, Craig.: *Embedded Linux,* Pearson Education, Indianapolis, USA, 2002, ISBN 0-672-32226-9.

[6] Yaghmour, Karim.: *Building embedded Linux systems,* O'Really & Assocites, Sebastopol, 2003, ISBN 0-596-00222-X.

[7] Kotzian J., Srovnal V.: *Can Based Distributed Control System Modelling Using UML.* In: Proceeding International Conference IEEE ICIT 2003, Maribor, Slovenia, ISBN 0-7803-7853-9, p.1012-1017.

[8] Kotzian J., Srovnal V.: *Development of Embedded Control System for Mobile Objects Using UML. In :* Programmable Devices and Systems  2004-IFAC Workshop, Krakow, Poland, IFAC WS 2004 0008 PL, ISBN 83-908409-8-7, p.293-298.

[9] Srovnal, V., Kotzian, J.: *FLYSYS–Flight Embedded Control System for Ultra-Light Airplane* . In proceeding 4th IFAC Symposium on Mechatronic Systems 2006, Duesseldorf, Germany, 2006, IFAC, p.998-1001.

[10] Kotzian J., Srovnal V. Jr,: *Distributed embedded system for ultralight airplane monitoring* , ICINCO 2007, Intelligent Control Systems and Optimization, Anger, France, 2007, ISBN 978-972-8865-82-5, p. 448-451.

[11] ONX Software Systems International Corporation: *QNX Neutrino RTOS–System Architecture* , Release V6.3 or later, Kanata, Ontario, Canada, 2007.

[12] Astle D., Durnil D.: *OpenGL ES Game Development*, Thomson Course Technology, Boston, MA, USA, 2006, ISBN 1-59200-370-2.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Towards Self-Managing Real-Time Systems

Roy Sterritt
University of Ulster
Jordanstown,
Northern Ireland
Email: r.sterritt@ulster.ac.uk

Mike Hinchey
Lero–the Irish Software Engineering Research
Centre, University of Limerick, Ireland
mike.hinchey@lero.ie.

***Abstract*—Upon first consideration, the Self-Managing Systems (and specifically Autonomic Systems) paradigm may seem to add too much additional functionality and overhead for systems that require strict real-time behavior. From our experience, not only does the Real-Time Systems area benefit from autonomicity, but the autonomic systems initiative requires the expertise of the RTS community in order to achieve its overarching vision. We briefly highlight some of the design considerations we have taken in incorporating self-management into Real-Time Systems.**

## I. Introduction

"AUTONOMIC Computing" is the term coined, initially by DARPA under the Autonomic Information Assurance program [13], and subsequently by IBM in their call to industry in 2001 [2]. The approach takes inspiration from the human and mammalian nervous systems in attempting to develop systems that are self-managing and ultimately self-governing [1].

We have argued in a previous article [3] that computer-based systems should be autonomic. We stand by that belief, in particular as software becomes more complex with greater expectations in terms of functionality, performance and real-time behavior. In our previous article [3], we were addressing a particular audience, viz. members of the IEEE Technical Committee on Engineering Computer-Based Systems (TC-ECBS). To that community, "computer-based systems" typically means embedded systems involving both hardware and software.

Such systems, like real-time systems, often have strict constraints in terms of performance and timing requirements. The addition of concepts from Autonomic Computing (AC) might at a first glance seem to add complexity rather than reduce it, something that real-time systems can generally ill-afford. We believe, however, that Autonomic Computing has much to offer in terms of reducing complexity, and our work over the last number of years has looked at how we might exploit AC and other biologically-inspired techniques in classes of systems that would otherwise be infeasible.

We believe that Autonomic Computing is not achievable without real-time systems (in particular at lower, implementation levels). Simultaneously, we believe that real-time systems can benefit much from AC, without significant overhead, and that future advancements necessitate that we move towards self-managing real-time systems.

## II. Self-Managing Systems

The vision of achieving self-management in our computing and communications systems has come to light under many similar initiatives—Autonomic Computing, Autonomic Communications, Organic Computing, Adaptive Infrastructure, N1, Dynamic System Initiative, Adaptive Network Care, Proactive Computing, Self-Organizing Systems, Self-*, Selfware, Biologically-Inspired Computing [5].

Although these different initiatives may have different academic specialties at their base, or were initiated by different industrial organizations, often they are inspired by self-management as exhibited in Biology and Nature. The IEEE has formalized the community as a Technical Committee on "Autonomous and Autonomic Systems" to underscore the strategic aims of developing next generation self- organizing and self-managing computing and communications infrastructures and systems.

### A. *Autonomic Computing*

The Autonomic Computing (AC) initiative focuses on managing complexity with self-managing systems, taking inspiration from the human autonomic nervous system (ANS) [1]-[5].

The ANS is that part of the nervous system that controls the vegetative functions of the body, such as circulation of the blood, intestinal activity, and secretion and production of chemical "messengers" (hormones) that circulate in the blood. The sympathetic nervous system (SyNS) supports "fight or flight", providing various protection mechanisms to ensure the safety and well-being of the body. The parasympathetic nervous system (PaNS) supports " rest and digest ", ensuring that the body performs necessary functions for long-term health.

The general properties of an autonomic (self- managing) system can be summarized by four objectives: being self-configuring, self-healing, self-optimizing and self-protecting, and four attributes: self-awareness, self-situated, self-monitoring and self-adjusting [7]. Essentially, the objectives represent broad system requirements, while the attributes identify basic implementation mechanisms.

Self-configuring represents a system's ability to re-adjust itself automatically; this may simply be in support of changing circumstances, or to assist in self-healing, self-optimization or self-protection.

Self-healing, in reactive mode, is a mechanism concerned with ensuring effective recovery when a fault occurs, identifying the fault, and then, where possible, repairing it. In proactive mode, it monitors vital signs in an attempt to predict and avoid "health" problems (reaching undesirable situations).

Self-optimization means that a system is aware of its ideal performance, can measure its current performance against that ideal, and has defined policies for attempting improvements. It may also react to policy changes within the system as indicated by the users. A self-protecting system will defend itself from accidental or malicious external attack. This necessitates awareness of potential threats and a means of handling those threats.

In achieving such self-managing objectives, a system must be aware of its internal state (self-aware) and current external operating conditions (self-situated). Changing circumstances are detected through self-monitoring and adaptations are made accordingly (self-adjusting).

As such, a system must have knowledge of its avail able resources, its components, their desired performance characteristics, their current status, and the status of inter-connections with other systems, along with rules and policies for how these may be adjusted. Such ability to operate in a heterogeneous environment will require the use of open standards to enable global understanding and communication with other systems.

These mechanisms are not independent entities. For instance, if an attack is successful, this will necessitate self-healing actions, and a mix of self-configuration and self-optimization, in the first instance to ensure dependability and continued operation of the system, and later to increase self-protection against similar future attacks. Finally, these self-mechanisms should ensure that there is minimal disruption to users, avoiding significant delays in processing.

The Autonomic Systems architecture essentially consists of cooperating autonomic elements made up of the component that is required to be managed, and the autonomic manager [11, 12]. It is assumed that an autonomic manager (AM) is responsible for a managed component (MC) within a self-contained autonomic element (AE). This autonomic manager may be designed as part of the component or provided externally to the component, as an agent for instance. To achieve a self-managing system these AEs will cooperate with remote autonomic managers through virtual, peer-to-peer, client-server or grid configurations.

### B. *Aims of AC*

If we consider the eight overriding objectives set out by IBM in 2001 in their call to the industry [2], some of these will seem, to the real-time systems (RTS) researcher or practitioner, characteristics that are common to real-time systems or that at least very desirable:

1. To be autonomic, a computing system needs to "know itself"—and comprise components that also possess a system identity.
2. An Autonomic Computing System must configure and reconfigure itself under varying and unpredictable conditions.
3. An Autonomic Computing system never settles for the status quo—it always looks for ways to optimize its workings.
4. An Autonomic Computing system must perform something akin to healing—it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction.
5. A virtual world is no less dangerous than the physical one, so an Autonomic Computing system must be an expert in self-protection.
6. An Autonomic Computing system knows its environment and the context surrounding its activity, and acts accordingly.
7. An Autonomic Computing system cannot exist in a hermetic environment.
8. *Perhaps* most critical for the user, an Autonomic Computing system will anticipate the optimized resources needed while keeping its complexity hidden.

This list of eight requirements, set out in [2] and [4], point to the fact that to qualify as an Autonomic System (AS), a system has to have a degree of self-awareness and familiarization with the components that compose it. It must know their capabilities, if it is to be able to adapt to its environment successfully, analogous to the way that dynamic scheduling requires a knowledge of tasks, their periods and deadlines.

An AS must be able to adapt to unpredictable conditions and adjust itself as necessary. This is something that is done, almost routinely, in an RTS. Real-Time Systems are constantly evolving. Their role is to adapt to changing environmental circumstances or new inputs. The role of scheduling is to optimize the use of resources and to make efficient use of available processing power. An effective RTS is performing this role.

A key focus of the list of essentials given above, is a move towards management and reduction of inherent complexity. Complexity is a major issue for any large-scale system. RTS have the added difficulty of having to meet strict timing-constraints and inputs from many sources, all of which need to be met.

But surely adding self-managing abilities to an already complex system, such as exhibited by an RTS, is only adding to adherent complexity? The overhead of self-management essentially adds an overhead to any system. Many RTSs are already pushing the limits of their schedules and adding any additional overhead is impossible.

It is our contention, however, that sometimes adding this overhead is justified. There are classes of systems for which self-management, and even self-government, is essential, for which timing constraints could not be met otherwise and for which the overhead of AC is more than repaid by a corresponding reduction in complexity.

### C. *Implementing AC*

It will be obvious to any practitioner that to hand over control to the system that was previously held by a human, in principle will add additional functionality, complexity and processing overhead to the system. Also experience from AI, Expert Systems, and Machine Learning problems will have shown many how slow decision-making systems can be. As such, it is vital to develop key principles and engineering techniques for how self-management systems should be created.

If we reconsider the meaning of the actual terms "Autonomic" and "Autonomous" [6]:

***au·to·nom·ic*** (àwtə nómmik)
*adj.*

1. Physiology.
   - a. Of, relating to, or controlled by the autonomic nervous system.
   - b. Occurring involuntarily; automatic: an autonomic reflex.
2. Resulting from internal stimuli; spontaneous.

***au·ton·o·mic·i·ty.*** (àwtə nóm i síttee)
*n.*

1. The state of being autonomic.

***au·ton·o·mous*** (aw tónnəməs)
*adj.*

1. Not controlled by others or by outside forces; independent: an autonomous judiciary; an autonomous division of a corporate conglomerate.
2. Independent in mind or judgment; self-directed.
3. 
   - a. Independent of the laws of another state or government; self-governing.
   - b. Of or relating to a self-governing entity: an autonomous legislature.
   - c. Self-governing with respect to local or internal affairs: an autonomous region of a country.
4. Autonomic.

[From Greek autonomos: *auto-*, auto- + nomos , *law*]

A difference stands out: "Autonomic" is very much concerned with spontaneous, reflex reactions while "Autonomous" is a slower, high level conscious decision-making process.

The basic principles of Autonomic and Autonomous systems can be incorporated into the design of a system to ensure that the correct response rate is achieved where it is needed. This has resulted in us considering a simple three tiered abstract architecture in our designs of self-managing systems:

- Autonomous Layer
- Selfware Layer
- Autonomic Layer

The *Autonomic Layer* is the bottom tier, closest to the hardware, and operates with immediate reaction to situations to ensure that system operations are maintained.

The *Selfware Layer* incorporates day-to-day operations of self-managing activity as and when needed, and as and when the system has the processing capacity available.

The *Autonomous Layer* is the top tier where high-level strategic objectives of the system are directed and satisfied over time. This often includes reflection.

As such, a key element that we have included in our work in designing AS is that of the *Autonomic Reflex,* borrowed from embedded and real-time systems and extended to include active system health telemetry.

### *D. Reflex Reactions*

Essentially, the aim of Autonomic C omputing is to create robust, dependable self-managing systems [8] in an attempt to deal with complexity.

At the heart of the architecture of any autonomic system are sensors and effectors. A control loop is created by monitoring behavior through sensors, comparing this with expectations (knowledge, as in historical and current data, rules and beliefs), planning what action is necessary (if any), and then executing that action through effectors. The closed loop of feedback control provides the basic backbone structure for each system component. There are two conceptual control loops in an Autonomic Element—one for self-awareness and another for self-situation (environmental awareness and context-awareness).

IBM represents this self-monitor/self-adjuster control loop as the monitor, analyze, plan and execute (MAPE) control loop. The monitor-and-analyze parts of the structure process information from the sensors to provide both self-awareness and an awareness of the external environment. The plan-and-execute parts decide on the necessary self-management behavior that will be executed through the effectors. The MAPE components use the correlations, rules, beliefs, expectations, histories, and other information known to the autonomic element, or available to it through the knowledge repository within the AM.

The autonomic environment requires that autonomic elements and, in particular, autonomic managers communicate with one another concerning self-* activities, in order to ensure th e robustness of the environment [9, 10]. It is our belief that the autonomic manager communications (AM ⇔ AM) must also include a reflex signal.

To facilitate this, fault-tolerant mechanisms such as a heart-beat monitor ('I am alive' signals) and pulse monitor (urgency/reflex signals) may be included within the autonomic element [9, 10]. See Figure 1 for an illustration of our autonomic environment.

The notion behind the pulse monitor (PBM) is to provide an early warning of an undesirable condition so that preparations can be made to handle the processing load of diagnosis and planning a response, including diversion of load. Together with other forms of communications it creates dynamics of autonomic responses [11] — the introduction of multiple loops of control, some slow and precise, others fast and possibly imprecise, fitting with the biological metaphors of reflex and healing [9].

This refle x component may be used to safeguard the autonomic element by communicating its health to another AE. The component may also be utilized to communicate environmental health information. For instance, in the situation where each PC in a LAN is equipped with an autonomic manager, rather than each of the individual PCs monitoring the same environment, a few PCs (likely the pillow queen—the least busy machines) may take on this role and alert the others via a change in pulse to indicate changing circumstances.

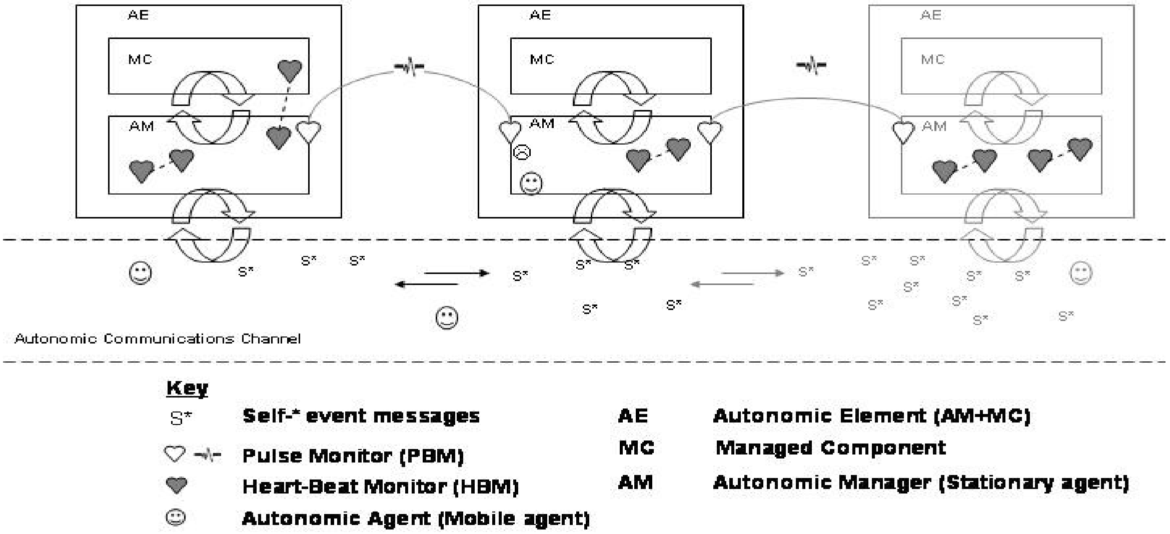

AE
MC
AM
S*
Autonomic Communications Channel
Key
S* Self-* event messages
Pulse Monitor (PBM)
Heart-Beat Monitor (HBM)
Autonomic Agent (Mobile agent)
AE Autonomic Element (AM+MC)
MC Managed Component
AM Autonomic Manager (Stationary agent)

Figure 1: An Autonomic Environment

An important aspect concerning the reflex reaction and the pulse monitor is the minimization of data sent—essentially only a "signal" is transmitted. Strictly speaking, this is not mandatory; more information may be sent, yet the additional information must not compromise the reflex reaction and the required real-time response of the system. For instance, in the absence of bandwidth concerns, information that can be acted upon quickly and not incur processing delays could be sent. The important aspect is that the information must be in a form that can be acted upon immediately and not involve processing delays (such as is the case of event correlation).

Just as the beat of the heart has a double beat (lu b-dub) the autonomic element's pulse monitor may have a double beat encoded as described above, a self health/urgency measure and an environment health/urgency measure. These match directly with the two control loops within the AE, and the self-awareness and environment awareness properties.

Another extension we have been working on is the adaptive pulse monitor; where the rate of the pulse adapts not only when alerting other self-managing components about concerns (increase in pulse rate) but to take into consideration bandwidth concerns. In particular, under fault conditions there is often a cascade and flood of event and alarm messages on the network, and we must actively reduce alerting so that achieving autonomicity is not making the situation worse.

## III. A Self-Managing RTS Example

### IV. *NASA Missions*

NASA missions require the use of complex hardware and software systems, and embedded systems, often with hard real -time requirements [3]. Most missions involve significant degrees of autonomous behavior, often over significant periods of time. There are missions which are intended only to survive for a short period, and others which will continue for decades, with periodic updates to both hardware and software. Some of these updates are pre-planned; others, such as with the Hubble Space Telescope, were not planned but now will be undertaken (with updates performed either by astronauts or via a robotic arm).

While missions typically have human monitors, many missions involve very little human intervention, and then often only in extreme circumstances. It has been argued that NASA systems should be autonomic [9, 14], and that all autonomous systems should be autonomic by necessity. Indeed, the trend is in that direction in forthcoming NASA missions.

We take as our example, a NASA concept mission, ANTS, which has been identified [15] as a prime example of an autonomic system.

#### 1) *ANTS*

ANTS is a concept mission that involves the use of intelligent swarms of spacecraft. From a suitable point in space (called a Lagrangian), 1000 small spacecraft will be launched towards the asteroid belt.

As many as 60% to 70% of these will be destroyed immediately on reaching the asteroid belt. Those that survive will coordinate into groups, under the control of a leader, which will make decisions for future investigations of particular asteroids based on the results returned to it by individual craft which are equipped with various types of instruments

*a) Self-configuring*

ANTS will continue to prospect thousands of asteroids per year with large but limited resources. It is estimated that there will be approximately one month of optimal science operations at each asteroid prospected. A full suite of scientific instruments will be deployed at each asteroid. ANTS resources will be configured and re-configured to support concurrent operations at hundreds of asteroids over a period of time.

The overall ANTS mission architecture calls for specialized spacecraft that support division of labor (rulers, messengers) and optimal operations by specialists (workers). A major feature of the architecture is support for cooperation among the spacecraft to achieve mission goals. The architecture supports swarm-level mission-directed behaviors, subswarm levels for regional coverage and resource-sharing, team/worker groups for coordinated science operations and individual autonomous behaviors. These organizational levels are not static but evolve and self-configure as the need arises. As asteroids of interest are identified, appropriate teams of spacecraft are configured to realize optimal science operations at the asteroids. When the science operations are completed, the team disperses for possible reconfiguration at another asteroid site. This process of configuring and reconfiguring continues throughout the life of the ANTS mission.

Reconfiguring may also be required as the result of a failure, such as the loss of, or damage to, a worker due to collision with an asteroid (in which case the role may be assumed by another worker, which will be allocated the task and resources of the original).

*b) Self-healing*

ANTS is self-healing not only in that it can recover from mistakes, but self-healing in that it can recover from failure, including damage from outside forces. In the case of ANTS, these are non-malicious sources: collision with an asteroid, or another spacecraft, etc.

ANTS mission self-healing scenarios span the range from negligible to severe. A negligible example would be where an instrument is damaged due to a collision or is malfunctioning. In such a scenario, the self-healing behavior would be the simple action of deleting the instrument from the list of functioning instruments. A severe example would arise when the team loses so many workers it can no longer conduct science operations. In this case, the self-healing behavior would include advising the mission control center and requesting the launch of replacement spacecraft, which would be incorporated into the team, which in turn would initiate necessary self-configuration and self-optimization.

Individual ANTS spacecraft will have self-healing capabilities also. For example, an individual may have the capability of detecting corrupted code (software), causing it to request a copy of the affected software from another individual in the team, enabling the corrupted spacecraft to restore itself to a known operational state.

*c) Self - optimizing*

Optimization of ANTS is performed at the individual level as well as at the system level.

Optimization at the ruler level is primarily through learning. Over time, rulers will collect data on different types of asteroids and will be able to determine which asteroids are of interest, and which are too difficult to orbit or collect data from. This provides optimization in that the system will not waste time on asteroids that are not of interest, or endanger spacecraft examining asteroids that are too dangerous to orbit.

Optimization for messengers is achieved through positioning, in that messengers may constantly adjust their positioning in order to provide reliable communications between rulers and workers, as well as with mission control back on Earth.

Optimization at the worker level is again achieved through learning, as workers may automatically skip over asteroids that it can determine will not be of interest.

*d) Self - protecting*

The significant causes of failure in ANTS will be collisions (with both asteroids and other spacecraft), and solar storms.

Collision avoidance through maneuvering is a major challenge for the ANTS mission, and is still under development. Clearly there will be opportunity for individual ANTS spacecraft to coordinate with other spacecraft to adjust their orbits and trajectories as appropriate. Avoiding asteroids is a more significant problem due to the highly dynamic trajectories of the objects in the asteroid belt. Significant planning will be required to avoid putting spacecraft in the path of asteroids and other spacecraft.

In addition, charged particles from solar storms could subject spacecraft to degradation of sensors and electronic components. The increased solar wind from solar storms could also affect the orbits and trajectories of the ANTS individuals and thereby could jeopardize the mission. One possible self-protection mechanism would involve a capability of the ruler to receive a warning message from the mission control center on Earth. An alternative mechanism would be to provide the ruler with a solar storm sensing capability through on-board, direct observation of the solar disk. When the ruler recognizes that a solar storm threat exists, the ruler would invoke its goal to protect the mission from harm from the effects of the solar storm, and issue instructions for each spacecraft to "fold" the solar sail (panel) is uses to charge its power sources.

*e) Self - aware*

Clearly, the above properties require the ANTS mission to be both aware of its environment and self -aware.

The system must be aware of the positions and trajectories of other spacecraft in the mission, of positions of asteroids and their trajectories, as well as of the status of instruments and solar sails.

### *V. Real Time Issues*

The swarm-based concepts of ANTS (or its submission, PAM—Prospecting Asteroid Mission—as described above) enable exploration missions that never before would be possible. Such concept missions are clearly real-time systems. ANTS must be survivable in a harsh environment (space) over multiple years. The mission must be able to protect itself and to recover from collisions, threats from solar storms, and other problematic issues.

This must all be considered in the context of significant transmission delays. Round-trip delays between Earth and the asteroid belt exceed 40 minutes. The result is that exceptional events cannot be dealt with from Earth. Even anticipated events cannot be dealt with from Earth, as catastrophic damage could have occurred before ground control had even received notification.

The result is that the system must be self-managing. In order for its real-time behavior to be realized, the mission must exhibit the properties of an Autonomic System, which (as we pointed out in Section II.A) are desirable properties of a RTS in any case.

## VI. Conclusion

What is clear, is that applications based on such paradigms as we have described, and many envisioned for the future (and certainly not limited to the aerospace or space exploration domain) will be far too complex for humans to address all issues. Moreover, many issues will not be foreseeable, and much behavior will require hard real-time deadlines that can never be met with more traditional approaches.

While there is an overhead to achieving autonomicity, we believe that this overhead comes with significant benefit for RTS. We do not believe that it is too "costly" 1 for Real-Time Systems, but rather than in the future it will prove to be essential for developing effective RTS. Simultaneously, Autonomic Computing, and related areas, draw on much of the excellent research produced by the RTS community, a significant proportion of which was essential in making the AC initiative feasible.

In short: we believe we are moving swiftly towards a time when it will be imperative to have self-managing Real-Time Systems.

## Acknowledgment

This work was supported in part by Science Foundation Ireland grant 03/CE2/I303_1 to Lero—the Irish Software Engineering Research Centre (www.lero.ie).

This research is partly supported at University of Ulster by the Computer Science Research Institute (CSRI) and the Centre for Software Process Technologies (CSPT) which is funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

Several of the technologies described in this paper were developed with NASA and are patent-pending and assigned to the United States Government.

## References

[1] M. G. Hinchey and R. Sterritt, "Self-Managing Software," *Computer,* 39(2):107-109, February 2006.

[2] P. Horn, "Autonomic computing: IBM perspective on the state of information technology," IBM T. J. Watson Labs, NY, 15th October 2001

[3] R. Sterritt, M. G. Hinchey, "Why Computer-Based Systems Should be Autonomic," *Proceedings of 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005),* Greenbelt, MD, USA, 3-8 April, 2005, Pages 406-414

[4] J. O. Kephart and D. M. Chess. "The vision of autonomic computing". *Computer ,* 36(1):41–52, 2003.

[5] R. Sterritt, "Autonomic Computing," Innovations in Systems and Software Engineering, Vol. 1, No. 1, Springer, ISSN 1614-5046, Pages 79-88 , 2005

[6] R. Sterritt, M. G. Hinchey, "Birds of a Feather Session: "Autonomic Computing: Panacea or Poppycock?", *Proceedings of IEEE Workshop on the Engineering of Autonomic Systems (EASe 2005) at 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005),* Greenbelt, MD, USA, 3-8 April, 2005, Pages 335-341

[7] R. Sterritt, D. W. Bustard, "Autonomic Computing: a Means of Achieving Dependability?" In *Proceedings of IEEE International Conference on the Engineering of Computer Based Systems (ECBS'03),* Huntsville, AL, USA, 7-11 April 2003, pp. 247-251.

[8] R. Sterritt, "Pulse Monitoring: Extending the Health-check for the Autonomic GRID," In *Proceedings of IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA 2003) at INDIN 2003,* Banff, AB, Canada, 22-23 August 2003, pp. 433-440.

[9] R. Sterritt, "Towards Autonomic Computing: Effective Event Management," In *Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW),* Maryland, USA, 3-5 December 2002, IEEE Computer Society Press, pp. 40-47.

[10] R. Sterritt, D. F. Bantz, "PAC-MEN: Personal Autonomic Computing Monitoring Environments," In *Proceedings of IEEE DEXA 2004 Workshops-2nd International Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 04),* Zaragoza, Spain, 30 August – 3 September, 2004.

[11] R. Sterritt and D. W. Bustard, "Towards an Autonomic Computing Environment," In *Proceedings of IEEE DEXA 2003 Workshops - 1st International Workshop on Autonomic Computing Systems,* Prague, Czech Republic, September 1-5, 2003, pp. 694-698.

[12] R. Sterritt and M. G. Hinchey, "From Here to Autonomicity: Self-Managing Agents and the Biological Metaphors that Inspire Them," *Proceedings of Integrated Design & Process Technology Symposium (IDPT 2005),* Beijing, China, 13-17 June, pp 143-150.

[13] S. M. Lewandowski, D. J. Van Hook, G. C. O'Leary, J. W. Haines, L. M. Rossey, "SARA: Survivable Autonomic Response Architecture," In Proc. DARPA Information Survivability Conference and Exposition II, Vol. 1, pp. 77-88, June 2001.

[14] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff, Autonomous and Autonomic Systems: A Paradigm for Future Space Exploration Missions, IEEE Trans. on Systems, Man and Cybernetics, Part C, 2006.

[15] W. F. Truszkowski, J. L. Rash, C. A. Rouff, and M. G. Hinchey, Asteroid Exploration with Autonomic Systems. *Proceedings 11th IEEE International Conference on Engineering Computer-Based Systems (ECBS), Workshop on Engineering Autonomic Systems (EASe),* Brno, Czech Republic, 24-27 May 2004, pp 484-489, IEEE Computer Society Press.

[16] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[17] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.

[18] *IEEE Criteria for Class IE Electric Systems* (Standards style), IEEE Standard 308, 1969.

[19] *Letter Symbols for Quantities*, ANSI Standard Y10.5-1968.

[20] R. E. Haskell and C. T. Case, "Transient signal propagation in lossless isotropic plasmas (Report style)," USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# IEC Structured Text programming of a small Distributed Control System

Dariusz Rzońca*, Jan Sadolewski*, Andrzej Stec*, Zbigniew Świder*, Bartosz Trybus*, and Leszek Trybus*
*Rzeszow University of Technology,
Faculty of Electrical and Computer Engineering
Wincentego Pola 2, 35-959 Rzeszów, Poland
Email: {drzonca, js, astec, swiderzb, btrybus, ltrybus}@prz-rzeszow.pl

***Abstract*—A prototype environment called CPDev for programming small distributed control-and-measurement systems in Structured Text (ST) language of IEC 61131-3 standard is presented. The environment consists of a compiler, simulator and configurer of hardware resources, including communications. Programming a mini-DCS (*Distributed Control System*) from LUMEL Zielona Góra is the first application of CPDev.**

## I. Introduction

DOMESTIC control-and-measurement industry manufactures transmitters, actuators, drives, PID (*Proportional-Integral-Derivative*) and PLC (*Programmable Logic*) controllers, recorders, etc. Connected into distributed systems, they are used for automation of small and medium scale plants. However, engineering tools used for programming such devices are rather simple and do not correspond to IEC 61131-3 standard [2] (Polish law since 2004).

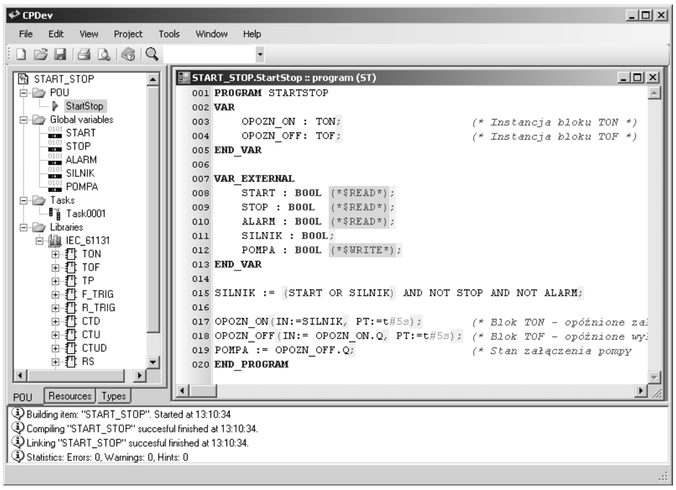
This paper presents current state of work on engineering environment called CPDev (*Control Program Developer*) for programming small control-and-measurement devices and distributed mini-systems according to the IEC standard (digits dropped for brevity). First implementation involves instruments from LUMEL Zielona Góra [7]. Initial information on CPDev was presented at the previous IMCSIT conference [5]. Similar environments have been described in [1], [6].

The CPDev environment (called also package) consists of three programs executed by PC and one by the controller. At the PC side we have:

- CPDev compiler of ST language,
- CPSim software simulator,
- CPCon configurer of hardware resources.

The programs exchange data through files in appropriate formats. The CPDev compiler generates an universal code executed by virtual machine VM at the controller side. The machine operates as an interpreter. The code is a list of primitive instructions of the virtual machine language called VMASM assembler. VMASM is not related to any particular processor, however it is close to somewhat extended typical assemblers. In this way, portability of the compiled code for different hardware platforms is provided. On the contrary, other solutions are usually built around the concept of translating IEC language programs into C code [6].

Basic characteristics of VMASM and virtual machine are given in [5]. CPSim simulator also involves the machine (in this case at the PC side).



```
PROGRAM STARTSTOP
VAR
    OPOZN_ON : TON;                        (* Instancja bloku TON *)
    OPOZN_OFF: TOF;                        (* Instancja bloku TOF *)
END_VAR

VAR_EXTERNAL
    START : BOOL (*$READ*);
    STOP : BOOL  (*$READ*);
    ALARM : BOOL (*$READ*);
    SILNIK : BOOL;
    POMPA : BOOL (*$WRITE*);
END_VAR

SILNIK := (START OR SILNIK) AND NOT STOP AND NOT ALARM;

OPOZN_ON(IN:=SILNIK, PT:=t#5s);        (* Blok TON - opóźnione za
OPOZN_OFF(IN:= OPOZN_ON.Q, PT:=t#5s); (* Blok TOF - opóźnione wy
POMPA := OPOZN_OFF.Q;                  (* Stan załączenia pompy
END_PROGRAM
```

```
Building item: "START_STOP". Started at 13:10:34
Compiling "START_STOP" succesful finished at 13:10:34.
Linking "START_STOP" succesful finished at 13:10:34.
Statistics: Errors: 0, Warnings: 0, Hints: 0
```

Fig. 1. User interface in CPDev environment

The CPDev package is developed in C# language of .NET Framework. The virtual machine is written in ANSI C and compiled with appriopriate, platform-dependent compilers e.g. avr-gcc in case of the LUMEL's mini-DCS. Other languages and programming environments are also used in specific cases.

The implementation of CPDev components was supported by lexical diagrams (compiler), object-oriented modelling techniques (programming environment) and coloured Petri nets (communication subsystem), see [5], [9].

## II. A program in ST language

Main window of user interface in the CPDev environment is shown in Fig. 1. It consists of three areas:

- tree of project structure, on the left,
- program in ST language, center,
- message list, bottom.

Tree of the START_STOP project shown in the figure includes Program Organization Unit (POU) with the program PRG_START_STOP, five global variables from START to PUMP, task TSK_START_STOP, and two standard function blocks TON and TOF from IEC_61131 library.

The PRG_START_STOP program seen in the main area is written according to ST language rules. The first part involves declarations of instances DELAY_ON, DELAY_OFF of the function blocks TON and TOF. Declarations of the global variables (EXTERNAL) are the second part, and four instructions of the program body, the third one. The instructions correspond to FBD (*Function Block Diagram*) shown in Fig. 2. So one can expect that certain MOTOR is turned on immediately after pressing a button START and a PUMP five seconds later. Pressing STOP or activation of an ALARM sensor triggers similar turn off sequence.

## III. Global variables and task

Global variables can be declared in CPDev either using individual windows or collectively at a variable list. The list for the START_STOP project is shown in Fig. 3.

The addresses specify *directly represented variables* [2], [3] and denote relative location in controller memory (keyword AT declares the address in individual window). Here these addresses are called *local*. Variables without addresses (not used in this project) are located automatically by the compiler.

Window with declaration of the TSK_START_STOP task is shown in Fig. 4. A task can be executed once, cyclically or continuously (triggered immediately after completing, as in small PLCs). There is no limit on the number of programs assigned to a task, however a program can be assigned only once.

Text of the project represented by the tree is kept in an XML text file. Compilation is executed by calling `Project->Build` from the main menu. Messages appear in the lower area of the interface display (Fig. 1). If there are no mistakes, the compiled project is stored in two files. The first one contains universal executable code in binary format for the virtual machine. The second one contains mnemonic code [5], together with some information for simulator and hardware configurer (variable names, etc.).

## IV. Functions and libraries

The CPDev compiler provides most of standard functions defined in IEC standard. Six groups of them followed by examples are listed below:

- type conversions: INT_TO_REAL, TIME_TO_DINT, TRUNC,
- numerical functions: ADD, SUB, MUL, DIV, SQRT, ABS, LN,
- Boolean and bit shift functions: AND, OR, NOT, SHL, ROR,

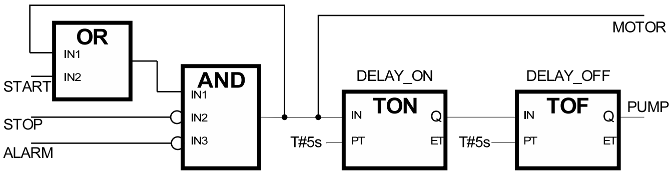
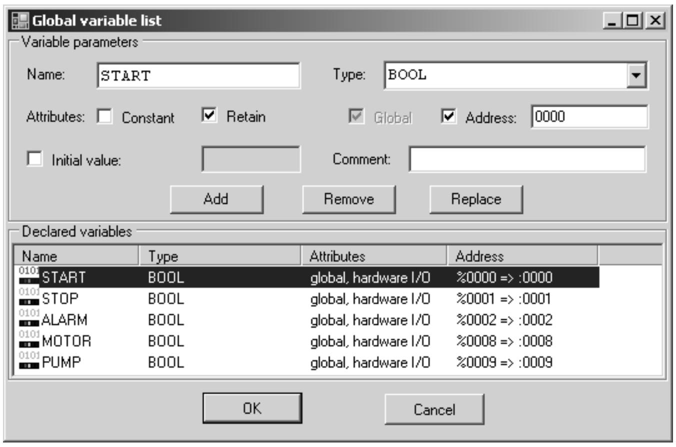


Fig. 2. START_STOP system for control of a motor and pump (with delay of 5 seconds)



Variable parameters

Name: START Type: BOOL

Attributes: □ Constant ☑ Retain ☑ Global ☑ Address: 0000

□ Initial value: Comment:

Declared variables

| Name | Type | Attributes | Address |
|---|---|---|---|
| START | BOOL | global, hardware I/O | %0000 => :0000 |
| STOP | BOOL | global, hardware I/O | %0001 => :0001 |
| ALARM | BOOL | global, hardware I/O | %0002 => :0002 |
| MOTOR | BOOL | global, hardware I/O | %0008 => :0008 |
| PUMP | BOOL | global, hardware I/O | %0009 => :0009 |

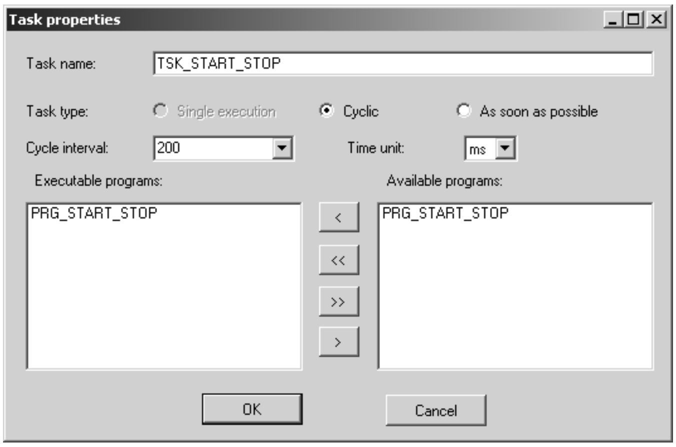
Fig. 3. Global variable list for the START_STOP project.

- selection and comparison functions: SEL, MAX, LIMIT, MUX, GE, EQ, LT,
- character string functions: LEN, LEFT, CONCAT, INSERT,
- functions of time data types: ADD, SUB, MUL, DIV (IEC uses the same names as for numerical functions).

Selector SEL, limiter LIMIT and multiplexer MUX from selection and comparison group are particularly useful. Variables of any numerical type, i.e. INT, DINT, UINT and REAL (called ANY_NUM in IEC [2], [3]) are arguments in most of relevant functions.

Typical program in ST language is a list of function block calls, where inputs to successive blocks are outputs from previous ones (see Fig. 2). So far the CPDev package provides three libraries:

- IEC_61131 standard library,
- Basic_blocks library with simple blocks supplementing the standard,
- Complex_blocks library for continuous regulation and sequential control.



Task name: TSK_START_STOP

Task type: ○ Single execution ● Cyclic ○ As soon as possible

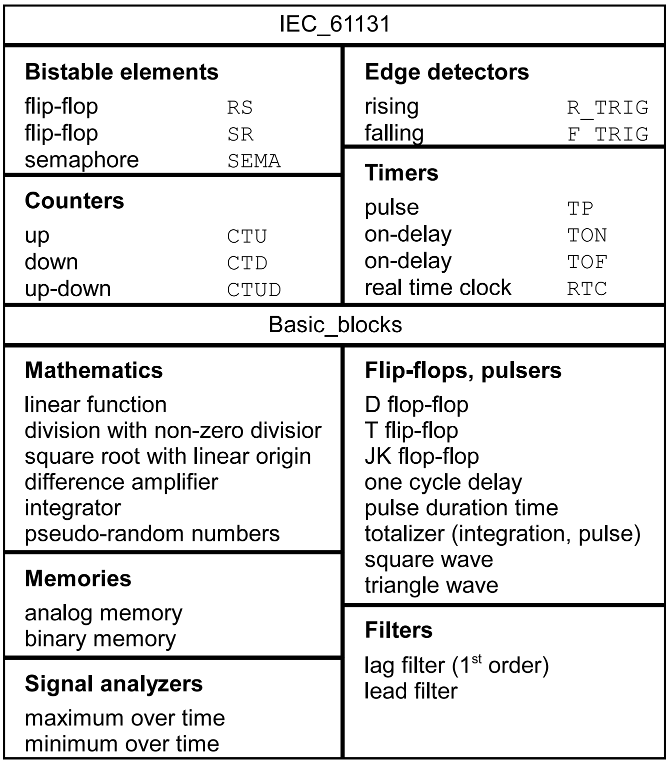
Cycle interval: 200 Time unit: ms

Executable programs: PRG_START_STOP

Available programs: PRG_START_STOP

Fig. 4. Declaration of TSK_START_STOP task

TABLE I
BLOCKS FROM IEC_61131 AND BASIC_BLOCKS LIBRARIES



| IEC_61131 | | | |
|---|---|---|---|
| **Bistable elements** | | **Edge detectors** | |
| flip-flop | RS | rising | R_TRIG |
| flip-flop | SR | falling | F_TRIG |
| semaphore | SEMA | **Timers** | |
| **Counters** | | pulse | TP |
| up | CTU | on-delay | TON |
| down | CTD | on-delay | TOF |
| up-down | CTUD | real time clock | RTC |
| **Basic_blocks** | | | |
| **Mathematics** | | **Flip-flops, pulsers** | |
| linear function | | D flop-flop | |
| division with non-zero divisior | | T flip-flop | |
| square root with linear origin | | JK flop-flop | |
| difference amplifier | | one cycle delay | |
| integrator | | pulse duration time | |
| pseudo-random numbers | | totalizer (integration, pulse) | |
| **Memories** | | square wave | |
| analog memory | | triangle wave | |
| binary memory | | **Filters** | |
| **Signal analyzers** | | lag filter (1$^{st}$ order) | |
| maximum over time | | lead filter | |
| minimum over time | | | |

Table I lists blocks form the first and second libraries. Blocks such as PID controller, servo positioner, multi-step sequencer, dosing block, etc., belong to the third library.

The user can develop functions, function blocks and programs, and store them in his libraries. Tables of single-size are available only.

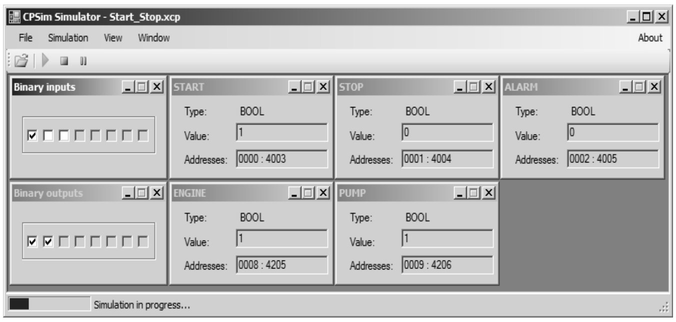
## V. CPSIM SIMULATOR

The compiled project may be verified by simulation before downloading into the controller. The CPSim simulator can be used in two ways:

- before configuration of hardware resources (simulation of the algorithm),
- after configuration of the resources (simulation of the whole system).

The first way involves logic layer of the CPDev environment. PC computer operates as a virtual machine executing universal code. The second way requires configuration of hardware resources, so it is application dependent. The CPCon configurer generates hardware allocation map (see below) that assigns local addresses to physical ones and specifies conversion of ST data formats into formats accepted by hardware. The objective is to bring simulation close to the hardware level, so CPSim uses both the code and the map. Simulation window of the START_STOP project is shown in Fig. 5. The two faceplates on the left present values of three inputs and two outputs (TRUE is marked). The user can select faceplates, arrange them on the screen and assign variables. Simulated values can be set both in group and in individual faceplates.

So far the window of Fig. 5 is used for simulation only. In future it will be also employed for on-line tests (*commissioning*).


CPSim Simulator - Start_Stop.xcp

Binary inputs: ☑ ☐ ☐ ☐ ☐ ☐ ☐ ☐

START — Type: BOOL — Value: 1 — Addresses: 0000 : 4003

STOP — Type: BOOL — Value: 0 — Addresses: 0001 : 4004

ALARM — Type: BOOL — Value: 0 — Addresses: 0002 : 4005

Binary outputs: ☑ ☑ ☐ ☐ ☐ ☐ ☐ ☐

ENGINE — Type: BOOL — Value: 1 — Addresses: 0008 : 4205

PUMP — Type: BOOL — Value: 1 — Addresses: 0009 : 4206

Simulation in progress...

Fig. 5. Simulation of the START_STOP project

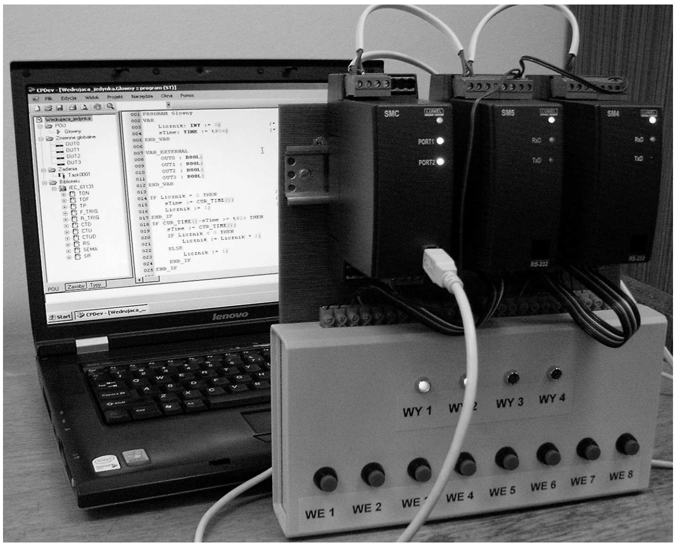


Fig. 6. Test set-up of mini-DCS system with SMC controller and SM I/O modules

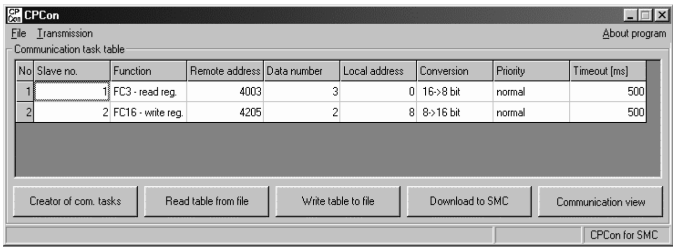
## VI. CPCON CONFIGURER AND MINI-DCS

The CPCon configurer defines hardware resources for particular application. The example considered here involves mini-DCS with SMC programmable controller, I/O modules of SM series and eventually other devices from LUMEL Zielona Góra [7]. Modbus RTU protocol is employed [4] on both sides of the SMC.

Fig. 6 shows test realization of the system with SMC controller (on the left), SM5 binary input module (middle), and SM4 binary output module (on the right). The console with pushbuttons and LEDs (below) is used for testing. The PC runs first the CPDev package and a SCADA (*Supervisory Control And Data Acquisition*) system later. PC and SMC are connected via USB channel configured as a virtual serial port.

The CPCon configurer functions are as follows:

- configuration of a communication between SMC and SM I/O modules,



Communication task table

| No | Slave no. | Function | Remote address | Data number | Local address | Conversion | Priority | Timeout [ms] |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | FC3 - read reg. | 4003 | 3 | 0 | 16->8 bit | normal | 500 |
| 2 | 2 | FC16 - write reg. | 4205 | 2 | 8 | 8->16 bit | normal | 500 |

Fig. 7. Communication configuration of the START_STOP project.

- creation of file with hardware resource allocation map,
- downloading the files with executable code and map to the SMC.

Recall that having the map the CPSim can be used in the second simulation mode.

Main window of the CPCon configurer is shown in Fig. 7. The Transmission slot sets speed, parity and stop bits for PC↔SMC and SMC↔SM communications. Communication task table determines what question↔answer and command↔acknowledgment transactions take place between SMC controller (*master*) and SM modules (*slaves*). The transactions are called *communication tasks* and represented by the rows of the table. The DCS system is configured by filling the rows, either directly in the table or interactively through a few windows of Creator of com. tasks (bottom).

The first row specifies communication between SMC and SM5 binary input module (remote). SM5 is connected to pushbuttons in the console (Fig. 6) which, in case of the START_STOP project, set the variables START, STOP and ALARM (Figs. 1, 2). In SMC these variables have consecutive addresses beginning from 0000 (Fig. 3). SM5 places the inputs in consecutive 16-bit registers beginning from 4003. So all variables can be read in a single Modbus transaction with the code FC3 (read group of registers [4]). However, since BOOL occupies single byte in CPDev, the interface of the virtual machine has to perform 16→8 bit conversion.

Communication tasks are handled by SMC during pauses that remain before end of the cycle, after execution of the program. Single transaction takes 10 to 30 ms, depending on speed (max. 115.2 kbit/s). If the pause is large, the task can be executed a few times. It has been assumed that the task with NORMAL priority is executed twice slower than the task with HIGH priority, and the task with LOW priority three times slower. As seen in Fig. 7, the communication with SM5 module has NORMAL priority. The Timeout within which transaction must be completed is 500 ms.

Second row of the Communication task table defines communication with the SM4 binary output module. SM4 controls the console LEDs. Two consecutive variables, MOTOR and PUMP, the first one with the local address 0008, are sent to SM4 by single message with the code FC16 to remote addresses beginning from 4205 (write group of registers). This time 8→16 bit conversion is needed.

## VII. Conclusions

CPDev environment for programming industrial controllers and other control-and-measurement devices according to IEC 61131-3 standard has been presented. The environment consists of ST language compiler, project simulator, and configurer of hardware resources, including communications. The user can program his own function blocks and create libraries. Mini-DCS control-and-measurement system form LUMEL is the first application of the package.

Programs written in the future in other IEC languages, first of all in FBD, will also be compiled to the VMASM code and executed by the virtual machine. Appriopriate compilers are under development. XML format for data exchange between languages has already been defined by PLCOpen [1], [8].

## Acknowledgment

Support from the MNiSzW R02 058 03 grant is gratefully acknowledged.

## References

[1] Bubacz P., Adamski M.: .NET platform and XML markup language in a software design system for logic controllers. *PAK*, 2006, no. 6bis, 94–96 (in Polish).

[2] IEC 61131-3 standard, *Programmable Controllers—Part 3, Programming Languages*. IEC, 2003.

[3] J. Kasprzyk: *Programming Industrial Controllers*. WNT, Warsaw, 2006 (in Polish).

[4] *Modicon MODBUS Protocol Reference Guide*. MODICON, Inc., Industrial Automation Systems, Massachusetts (1996) http://www.modbus.org/docs/PI_MBUS_300.pdf

[5] D. Rzońca, J. Sadolewski, B. Trybus: Prototype environment for controller programming in the IEC 61131-3 ST language. *Computer Science and Information Systems*, December 2007 (also 2007 IMCSIT, 1041 – 1054).

[6] Tisserant E., Bessard L., de Sousa M.: An Open Source IEC 61131-3 Integrated Development Environment. $5^{th}$ *Int. Conf. Industrial Informatics*, Piscataway, NJ, USA, 2007.

[7] http://www.lumel.com.pl

[8] XML Formats for IEC 61131-3 ver. 1.01 – Official Release. http://www.plcopen.org/

[9] Rzońca D., Trybus B.: Timed CPN model of SMC controller communication subsystem, in: S. Węgrzyn, T. Czachïż¡rski, A. Kwiecień (Eds.): *Contemporary Aspects of Computer Networks*, WKŁ, Warszawa 2008, 203–212.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Web-based Laboratories: How Does That Affect Pedagogy? Thoughts for a Panel Discussion

Janusz Zalewski
Department of Computer Science
Florida Gulf Coast University
Fort Myers, FL 33965-6565, USA
zalewski@fgcu.edu

***Abstract* — The author presents a few questions for discussion at the Panel on web-based laboratories. The key issue is whether and how such laboratories can enhance the pedagogy. Is this only a technological trend or are there real values in creating online labs for real-time systems education? Answers to sample questions from the literature are reviewed.**

## I. Introduction

WEB-BASED or online laboratories are a very hot issue in contemporary education in all engineering and science disciplines. They are considered to be the technology alternative and complementary to traditionally organized hands-on labs, where students have direct physical contact with equipment to conduct experiments, as well as to simulation labs, where experiments are modeled using computer software and access is provided either directly in the lab or virtually via the Internet.

While web-based labs are a very attractive option to pursue, especially in cases with expensive lab equipment, when only a few institutions can afford it, or in cases when non-traditional students have difficulties meeting the time frame the labs are offered, there are still several issues related to pedagogy, whether lab experiments conducted via the web have intrinsic value and provide the level of education and experiences comparable with traditionally conducted labs. These issues are of concern to faculty teaching courses with such labs as well as to the institutions hosting these labs, since the investment of time and resources in creating and maintaining web-based labs is significant and needs to be justified in a longer term.

The objective of this panel session is to review some of the existing approaches to offering web-based laboratories and discuss issues affecting the pedagogy of such labs. The discussion is based on the most recent publications on web-based labs reviewed by the panel proponent. A summary of the issues related to pedagogy raised in some of these publications is presented below.

## II. Questions Related to Pedagogy

Serious publications on web-based laboratories began to appear in the early 2000's [1]-[3]. They represented an overall trend in transitioning to intercontinental learning [4]-[5] and concerned a variety of disciplines, from sciences [6] to engineering [7]-[9]. More recently, a proliferation of papers on remote, web-based labs is observed. They report on developments not only in disciplines traditionally considered the most advanced in remote experimentation, such as control engineering and robotics [10]-[12], but also in electronics, in courses using FPGA's [13] and PLC's [14], optical circuits [15] and antennas [16], as well as in physics [17], chemical engineering [18], materials research [19] and energy research [20]. Some authors propose generic labs [21], as well as try to analyze the phenomenon in comparison to hands-on and simulation labs [22]. No major work however, to the author's knowledge, has been done in bringing computer science or software engineering labs to the web.

### A. Four Basic Questions

In this view, the author believes that some basic questions need to be asked, first, regarding the existence or necessity of building such labs. Assuming the basic architecture of a web-based lab, as illustrated in Fig. 1, the following four questions come to mind:

Q1) What is the place of a web-based lab in a Computer Science or a Software Engineering program or a course?

Q2) Once the lab is in place, for conducting computer science or software engineering experiments in a web based lab: is there an ideal model?

Q3) Developing software for remote equipment: how does a web-based lab help?

Q4) Is there really an intellectual value added to a course with remote web-based laboratories?

My understanding of these four questions is as follows: Q1 is setting the stage for the discussion, formulates the requirements for a lab, Q2 and Q3 are meant to discuss the practice of remote labs, and Q4 is meant to stimulate the participants to derive some conclusions.

This work was supported in part by a grant from the National Science Foundation, Award No. DUE-0632729. Findings and opinions contained herein are not necessarily those of the NSF.

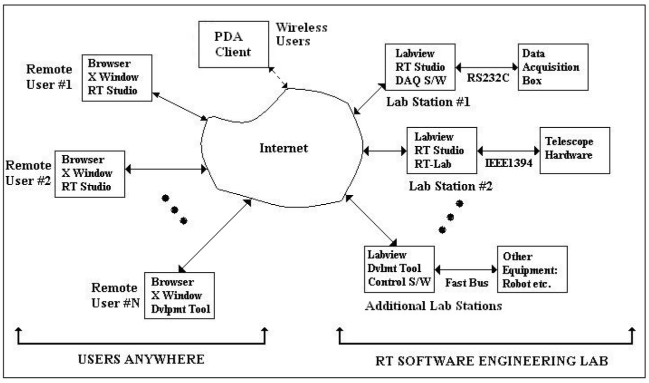


Fig. 1 System architecture of a typical web-based lab.

### B. Addressing the Pedagogy

Educators who have already implemented web-based labs often present their views on educational values of such labs and the challenges they are facing. Ma and Nickerson [22] review a number of early decade publications on web-based labs and mention several advantages such labs might bring to the table:

- sharing expensive experimental devices with a pool of schools
- increasing the availability of devices over time and their accessibility from different locations
- increasing the number of students that have lab access
- increasing student motivation and interest in conducting experiments.

Schools that started building and offering online labs first, offer a number of interesting observations what such labs can bring to pedagogy [21]:

- web labs enhanced conceptual learning, stimulated higher-order thinking, and reinforced individual styles of learning in multiple ways
- web labs allowed students to control their own learning process, while enabling faculty to maintain factual rigor and coherence throughout the course.

Among the challenges, ones the most frequently listed include: providing extended technical support, merit-based assistance to students in extended hours, and practical cost to maintain such labs.

## III. Summary and Conclusion

Being an important new lab component of any course, in addition to hands-on and simulation labs, web-based labs are still facing multiple challenges. It is an open question whether they increase an educational value of a course. It is also unclear whether they have a chance one day to replace traditional labs, filled with instrumentation and operated like a shop. One thing remains certain, however, as the world of technology clearly moves towards operating devices remotely, student experience with remote control of equipment will become invaluable on the job. It is hoped that some of the questions asked in this note will be answered in the discussion.

## References

[1] Gillet D. et al., Recent Advances in Remote Experimentation, *Proc. 2000 American Control Conf.*, Chicago, Ill., June 2000, Vol. 4, pp. 2955-2956

[2] Joler M., C.G. Christodoulou, Laboratories Accessible through the Internet, *IEEE Microwave Magazine*, Vol. 2, No. 4, pp. 99-103, December 2001

[3] Grushow A., A.J. Brandolini, NMR Spectroscopy: Learned and Delivered on the Internet, *Chemical Educator*, Vol. 6, pp. 311-312, 2001

[4] Shah A. et al., Web-based Course on Software Quality Assurance: Perspectives on Intercontinental Learning, *Proc. ICEE'99, International Conference on Engineering Education*, Ostrava-Prague, Czech Republic, August 10-14, 1999

[5] Thiriet J.-M., Toward a Pan-European Virtual University in Electrical and Electrical Information Engineering, *IEEE Trans. on Education*, Vol. 45, No. 2, pp. 152-160, May 2002

[6] Lacy C.H.S, Observational Research for All Students, *Astronomy Education Review*, Vol. 2, No. 2, pp. 129-137, 2003

[7] Watson J. L. et al., On-line Laboratories for Undergraduate Distance Engineering Students, *Proc. 34th ASEE/IEEE Frontiers in Education Conference*, Savannah, GA, October 20-23, 2004, pp. TC3-1/6

[8] Casini M., D. Prattichizzo, A. Vicino, The Automatic Control Telelab: A Web-based Technology for Distance Learning, *IEEE Control Systems*, Vol. 24, No. 3, pp. 36-44, June 2004

[9] Duan B. et al., An Online Laboratory Framework for Control Engineering Courses, *Int. Journal on Engineering Education*, Vol. 21, No. 6, pp. 1068-1075, 2005

[10] Payá L. et al., Distributed Platform for the Control of the Wifibot Robot through the Internet, *Proc. 7th IFAC Symp. on Advances in Control Education* , Madrid, June 21-23, 2006

[11] Marangé P., F. Gellot, B. Riera, Remote Control of Automation Systems for DES Courses, *IEEE Trans. on Industrial Electronics*, Vol. 54, No. 6, pp. 3103-3111, December 2007

[12] Hovland G., Evaluation of an Online Inverted Pendulum Control Experiment, *IEEE Trans. on Education*, Vol. 51, No. 1, pp. 114-122, February 2008

[13] Datta K., R. Sass, Rboot: Software Infrastructure for a Remote FPGA Laboratory, *Proc. FCCM 2007, 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, Calif., April 23-25, 2007

[14] Marques R. et al., Design and Implementation of a Reconfigurable Remote Laboratory, Using Oscilloscope/PLC Network for WWW Access, *IEEE Trans. on Industrial Electronics,* Vol. 55, No. 6, pp. 2425-2432, June 2008

[15] Gurkan D., A. Mickelson, D. Benhaddou, Remote Laboratories for Optical Circuits, *IEEE Trans. on Education*, Vol. 51, No. 1, pp. 53-60, February 2008

[16] Stancil D.D., N. Gist, Y. Jiang, REAL: The Remote Educational Antenna Laboratory, *Proc. IEEE International Symposium on Antennas and Propagation*, June 2007, pp. 5399-5402

[17] Maziewski A., W. Dobrogowski, V. Zablotskii, GloLab: Creating a Global Internet-accessible Laboratory, *Physics Education*, Vol. 42, No. 1, pp. 72-75, January 2007

[18] Guo J., D. Kettler, M. Al-Dahhan, A Chemical Engineering Laboratory over Distributed Control and Measurement Systems, *Computer Applications in Engineering Education*, Vol. 15, No. 2, pp. 174-184, 2007

[19] Genis A. et al., Development of NDE Laboratory for AET Students and Certification Program, *Proc. 2007 ASEE Annual Conf.*, Honoloulou, Hawaii, June 24-27, 2007, Paper AC-2007-251

[20] Pedersen K.O.H. et al., Wind Turbine Measurement Technique – an Open Laboratory for Educational Purposes, *Wind Energy*, Vol. 11, No. 3, pp. 281-295, 2008

[21] Harward V.J. et al., The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories, *Proceedings of the IEEE*, Vol. 96, No. 6, pp. 931-950, June 2008

[22] Ma J., J. V. Nickerson, Hands-on, Simulated, and Remote Laboratories: A Comparative Literature Review, *ACM Computing Surveys*, Vol. 38, No. 3, Article #7, 2006

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 763–771

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Improving Energy-Efficient Real-Time Scheduling by Exploiting Code Instrumentation

Thorsten Zitterell and Christoph Scholl
Albert-Ludwigs-University, Freiburg im Breisgau, Germany
{tzittere|scholl}@informatik.uni-freiburg.de

***Abstract*—Dynamic Frequency and Voltage Scaling is a promising technique to save energy in real-time systems. In this work we present a novel light-weight energy-efficient EDF scheduler designed for processors with discrete frequencies which performs on-line intra- and inter-task frequency scaling at the same time. An intra-task scheduling scheme based on cycle counters of a processor allows the application of our approach to shared code of library functions and to task setups where only sparse intra-task information is available. Our ‘Intra-Task Characteristics Aware EDF’ (ItcaEDF) scheduler which aims to run with a low frequency by eliminating idle time and inter- and intra-task slack times was evaluated in an compiler, WCET analysis, and simulation framework. Our experiments show that state-of-the-art intra-task as well as inter-task frequency scaling approaches are clearly outperformed by our approach.**

## I. Introduction

REAL-TIME systems like mobile multimedia devices have to meet high demands which are often competing. They require high performance under real-time constraints (for audio en-/decoding, e.g.), but also have to deal with limited resources like energy. Dynamic Voltage and Frequency Scaling (DVS/DFS) is an effective instrument to control the trade-off between system performance and energy-efficiency. Here, the operating system or the applications can decide whether the system should run at a higher frequency for higher performance but also with more power consumption – or at a lower frequency to save energy. When the timing behavior of real-time systems is specified by deadlines for individual tasks in the system, DFS (Dynamic Frequency Scaling) algorithms try to decrease the processor frequency under the constraint that deadlines are still guaranteed.

The concept of Dynamic Voltage and Frequency Scaling which changes the supply voltage and the processor frequency during run time has been proposed in many publications, e.g., [1], [2], [3], [4], [5], [6], [7], as a technique to reduce energy consumption for systems with and without real-time constraints. Existing approaches to DVS can be divided into inter-task voltage scheduling and intra-task voltage scheduling.

Approaches to *inter-task* voltage scaling make use of the fact that computation times of tasks are usually not fixed due to different execution paths during run time. Whereas standard scheduling approaches for real-time systems are based on fixed worst-case execution times for the tasks, inter-task voltage scaling makes use of ‘inter-task slack times’, which occur when a task instance terminates earlier than expected for the worst-case. For instance, Lee and Shin [8] adjusted the EDF scheduler [9] to the voltage scaling context. They presented an energy-efficient on-line EDF algorithm which requires only constant time for each context switch. In their approach voltage scaling is only performed when a task starts or resumes after another task which leaves some ‘inter-task slack time’ to the following task. [8] assumes a processor model which is able to provide a continuous spectrum of frequencies (limited by a maximum frequency $f_{max}$).

Whereas in many cases inter-task voltage scaling approaches succeed in reducing energy consumption, there are application scenarios where inter-task voltage scaling is less effective: Since voltage scaling can be applied only between two activations of different tasks, the granularity of inter-task voltage scaling may not be high enough, especially when the system consists of a small number of relatively long-running tasks (with one task as an extreme case).

In contrast to inter-task voltage scaling, *intra-task* voltage scaling algorithms perform voltage scaling during the execution of single tasks (and not between two activations of different tasks). Typically, there are many points in the program for a single task where frequency scaling is performed in case that execution time was saved (relative to the worst-case execution time). In that way, slack time for downsizing the processor frequency can be used much earlier than by inter-task voltage scaling and at a higher granularity. Compared to inter-task methods this may potentially lead to a more homogeneous distribution of used processor frequencies and thus to higher energy savings.[1] In [10] control flow analysis determines execution paths in a program whose traversal will save cycles and result in earlier task termination. Depending on the number of saved cycles the executable code is instrumented at *Voltage Scaling Edges (VSEs)*. Voltage scaling edges are restricted to branches in the control flow. At each edge the processor frequency is decreased by a given factor and the system can save energy. The basic concept has been further optimized by identifying earlier voltage scaling points using data flow information of the program [11] and by using profile information to estimate probabilities of branches [12].

[1] Idle times, which indicate a non-optimal voltage scaling strategy, can not always be avoided by inter-task methods: In the extreme case, the slack produced by a task running much shorter than the worst case execution time can not be used by other tasks, since no instances of other tasks have been released at the finishing time of that task (even if processor utilization is maximal assuming worst-case timing).

The intra-task algorithms mentioned above concentrate on single-task environments and they consider processors with a continuous frequency range. Thus, these algorithms are able to enforce that a task's execution completes exactly at its deadline, provided that there is a frequency scaling point at each branch in the control flow. However, this is not a realistic scenario:

- Usually, processors do not offer a continuous frequency range, but a set of discrete frequencies. On processors with discrete frequencies the frequency required by the intra-task algorithm may not be available and the processor has to run at some higher speed, thus the task may finish *before* its deadline.
- Moreover, in real applications it does not make sense to place a frequency scaling point at *every* branch in the control flow because of the overhead both for frequency scaling itself and for computing saved cycles and scaling factors at run time. For that reason an additional slack may occur.
- Finally, there may also be other reasons for the situation that a task is not able to keep account of every cycle it has saved compared to worst-case estimations during worst-case execution time (WCET) analysis: Especially for more complex architectures including caches and pipelines the execution times really needed may be smaller than the WCET assumptions, though tasks fail to detect these saved cycles at frequency scaling points.

Based on these observations the idea of our approach was to combine the advantages of inter-task and intra-task frequency scaling in an on-line real-time scheduler. More precisely, we propose a multi-level approach to DFS for a set of periodic tasks: The first level tries to eliminate idle times in case that the processor is not fully utilized. The optimal frequency for the task set is dynamically recalculated whenever tasks are created or removed. Whereas the first level is only based on worst-case execution times, the second and the third level take into account that the actual computation times are often much lower that the worst-case times: At level two we perform intra-task frequency scaling for each task individually. Finally, slack times produced by intra-task frequency scaling are used by inter-task frequency scaling at level three.

The idea of combining intra- and inter-task frequency scheduling has already been previously followed by Seo et al. [13], but in a completely different context: Seo et al. consider a fixed and finite set of aperiodic tasks (potentially including task dependencies) and they compute an optimal schedule using an off-line algorithm. Preemption is not allowed in their model, i.e., the schedule consists of starting times $s_i$ and ending times $e_i$ for each task $\tau_i$. Intra-task scheduling is then based on branching probabilities and for a task $\tau_i$ it computes (under the assumption of a continuous frequency range) a frequency distribution which minimizes the average case energy consumption provided that the execution time will be exactly $e_i - s_i$.

In contrast, our approach handles *dynamic* sets of *periodic* tasks which are scheduled by an *on-line* algorithm. Moreover, our scheduler allows *preemption* and voltage scaling is performed for processors with a *discrete set of frequencies*. The contributions of our paper are as follows:

- We make intra-task frequency scaling applicable in a multi-task environment and we make use of it in a tight interaction with inter-task scheduling. We provide an online algorithm suitable for dynamically changing sets of periodic tasks and we support preemptive task execution.
- Thereby, we provide a simplified scheme for computing information necessary for rescaling in the intra-task context. This scheme is based on a cycle counter in the processor hardware and it makes it possible to directly instrument different execution paths within (nested) loops.
- Our scheme for code instrumentation is also suitable for shared code in libraries used by different tasks.
- We provide a method based on a realistic processor model with a set of discrete frequencies instead of a continuous range of frequencies. The method can directly be integrated into an operating system if overhead for context switches and for frequency scaling is considered according to Section III-F.

The paper is organized as follows. We first give some preliminaries and definitions for inter- and intra-task real-time scheduling in Section II. Section III presents the concepts and implementation of our three-level approach. Experimental results are described in Section IV. Finally, Section V concludes the paper.

## II. Preliminaries

### A. Processor and Hardware Model

The processor model used in this paper provides $m$ different operating configurations $S = \{s_1, ..., s_m\}$ with $s_i = (f_i, v_i)$. Each pair $(f_i, v_i)$ consists of a frequency $f_i$ and a voltage $v_i$ with $f_i < f_j$ for $1 \leq i < j \leq m$. Therefore, the minimum and maximum possible frequencies are $f_1$ and $f_m$, respectively. The hardware also provides a cycle counter $\check{C}_{CLK}$ which is automatically incremented during task execution and a frequency-independent clock providing the real time $t$.

### B. Energy Model

In order to evaluate the energy-efficiency of our scheduler, we use the assumptions that the power consumption in CMOS circuits scales quadratically to the supply voltage ($P \propto f \cdot v^2$) [14]. Therefore, we compute the (dynamic) energy consumption $E$ of the system by $E = const \cdot \sum_{s_i \in S} c_{f_i} \cdot v_i^2$ provided that $c_{f_i}$ cycles have been executed at frequency $f_i$ or voltage $v_i$, respectively.

### C. Earliest Deadline First

An Earliest Deadline First (EDF) scheduler [9] always executes the task with the earliest deadline. A task $\tau_i$ is specified by its computation cycles $\check{C}_i$ and its period $T_i$. The symbol ˇ will be used to denote variables containing cycle-based values. Thus, a task set is specified by $\Gamma = \{\tau_i(\check{C}_i, T_i), i = 1, .., n\}$.

```
 1: while cond do
 2:    // max. 3 iterations
 3:    if cond1 then
 4:       $b_1$;
 5:    end if
 6:    $b_2$;
 7: end while
 8: if cond2 then
 9:    $b_3$;
10: end if
11: $b_4$;
```

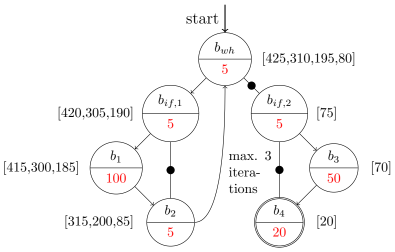


Fig. 1. Example code and corresponding Control Flow Graph (CFG) with timing information: The nodes represent basic blocks including their execution cycles. The values in brackets next to each node give the remaining worst-case execution cycles for each loop iteration.

We assume that the relative deadline $D_i$ of a task is the same as the period $T_i$, the absolute deadline $d_i$ for a task instance increments with $T_i$ for each instance.

In real-time systems, the worst-case execution cycles $\check{C}_i$ of a task $\tau_i$ for a specific processor *internally* depends on the task code and can be determined by static code analysis, for example. However, the period $T_i$ of a task is *externally* specified by the system designer for a task function. Moreover, the worst-case computation time $C_i$ of a task $\tau_i$ can be *derived* from the computation cycles $\check{C}_i$ if the execution frequency $f$ of the processor is known: $C_i = \check{C}_i \cdot f^{-1}$. The worst-case finishing (ending) time is $e_i$, e.g., a task which is activated at $t_{start}$ has an absolute completion time $e_i = t_{start} + C_i$ if it is not preempted.

The utilization factor $U$ of a processor under a given set $\Gamma$ of $n$ independent periodic tasks is defined as $U = \sum_{i=1}^{n} \frac{C_i}{T_i}$. Such a task set is schedulable with EDF if and only if $U \leq 1$.

### D. Intra-task scheduling

Intra-task scheduling aims to save power by lowering the processor frequency and simultaneously by eliminating internal task slack times. Such slack times always occur whenever there is a branching in the control flow of a task and the actual execution path involves fewer cycles than the worst-case execution path. In the following, we will give a brief overview of intra-task voltage scaling based on [10].

Consider the example code and corresponding *Control Flow Graph* (CFG) in Figure 1. Each node represents a basic block. The nodes include the number of cycles for the execution of the corresponding blocks. The values in brackets indicate the remaining worst-case execution cycles when entering a specific block. Multiple values refer to nodes which can be executed several times, i.e., which are part of a loop.

In the case that condition *cond* is not true, block $b_{if,2}$ will be directly executed after block $b_{wh}$. If this situation occurs before any execution of the `while`-loop, the remaining worst-case execution cycles ($\check{C}_{RWEC}$) after block $b_{wh}$ decrease from $\check{C}_{RWEC}(b_{if,1}) = 425 - 5 = 420$ to $\check{C}_{RWEC}(b_{if,2}) = 75$ cycles. The frequency within the task can be reduced by the scaling factor $\beta(b_{wh} \rightarrow b_{if,2}) = \frac{75}{420}$, because the current frequency was chosen in a way that the execution of 420 cycles will be finished in time, but only 75 cycles are needed. Now the processor frequency $S(b_{if,2})$ at block $b_{if,2}$ is updated according to $S(b_{if,2}) = S(b_{wh}) \cdot \beta(b_{wh} \rightarrow b_{if,2})$.

However, this scaling factor changes, if the `while`-loop is executed 1,2 or at most 3 times. As this information is not known in advance, the authors in [10] use so-called L-Type Voltage Scaling Edges (VSE) which calculate the scaling factor $\beta(b_{wh} \rightarrow b_{if,2})$ during run time depending on the number of loop iterations.

A second type are B-Type VSEs and they are used for `if`-statements (line 8 in our example code). As the number of saved cycles does not depend on loop iterations the scaling factor is $\beta(b_{if,2} \rightarrow b_4) = \frac{20}{70}$. Here, we only need 20 cycles, but remaining 70 cycles were in the budget of the task. Note that this simple scheme for B-Type VSEs can not be used, if we consider branches within loops (line 3, e.g.). In that case the number of remaining cycles depends on the iteration count of the outer loop and the scaling factor $\beta(b_{if,1} \rightarrow b_2)$ has to be dynamically computed at run time. (The scaling factor is equal to $\frac{315}{415}$, $\frac{200}{300}$, or $\frac{85}{185}$, depending on whether the loop is iterated for the first, second, or third time). Of course, the computation of the right scaling factor becomes even more involved if the branch is within a larger number of nested loops.

In contrast, our scheduler accepts relative information about the number of saved cycles in order to simplify frequency scaling. As we will describe in more detail in Section III-B1a, our scheme not only simplifies the instrumentation of code, but also makes instrumentation of shared code possible. This scheme is used in an overall approach which integrates intra-task frequency scaling and inter-task frequency scaling for preemptive environments with a dynamic number of tasks and a processor model with discrete frequencies.

## III. Energy-Efficient Intra- and Inter-task Frequency Scaling

Our method for real-time scheduling which is presented in this section aims at obtaining a homogeneous distribution of processor frequencies at a level which is as low as possible and thus minimizes energy consumption. In order to guarantee temporal constraints, the utilization of idle times, intra- and inter-task slack times has to be performed without violating deadlines of tasks.

### A. Idle-time Elimination

For EDF, a real-time schedule contains idle times iff $U < 1$. These idle times can be minimized if a processor is run with a lower frequency $f_\alpha$ (and indirectly with lower voltage and power consumption). Schedulability is still guaranteed when executing all $n$ tasks with a frequency $f_\alpha$ according to

$$\sum_{i=1}^{n} \frac{\check{C}_i / f_\alpha}{T_i} = 1 \Leftrightarrow f_\alpha = \sum_{i=1}^{n} \frac{\check{C}_i}{T_i}.$$

The frequency $f_\alpha$ is the lowest frequency to run the processor without violating real-time constraints provided that computation cycles $\check{C}_i$ are fixed to their worst-case estimation. However, computation cycles of task execution are not constant – they can be reduced by branching conditions or premature abort of loops during run time. In that case we can decrease the processor frequency even more in order to execute less cycles in the same time according to the intra- and inter-task slack-time elimination scheme as described below.

### B. Intra-Task Slack-time Elimination

In Section II-D, we have given a brief introduction to intra-task frequency scaling. In our work the intra-task frequency scaling is supported by the operating system which takes slack times into account for processors with discrete frequencies. Our approach differs from [10] with respect to three issues:

- We use a simplified scheme for computing remaining worst-case execution cycles. This scheme is based on an estimation of worst-case execution cycles for the task as a whole (by static analysis), relative information about saved execution cycles within the control flow, and a cycle counter of the processor. This scheme has the additional advantage that our method is suitable for shared code in libraries used by different tasks.
- The computation of frequencies is not based on scaling factors which take into account the previous estimation for worst-case remaining cycles and a new estimation due to saved cycles (as in [10]), but it is changed in order to be able to make use of inter-task slack time and of intra-task slack time which remain due to a discrete set of processor frequencies.
- The computation of worst-case remaining execution cycles $\check{R}_i$ of a task $\tau_i$, the computation of new processor frequencies, and the according change of frequencies can be seamlessly integrated into an operating system.

*1) Keeping track of remaining worst-case execution cycles:* As a basis for frequency rescaling we need an estimation of worst-case remaining execution cycles $\check{R}_i$ of tasks $\tau_i$. As described in Sect. II-D the computation of worst-case remaining execution cycles $\check{R}_i$ or the scaling factor, respectively, as given in [10] may need rather involved computations at run time. Especially if a frequency scaling point is located in several nested loops, an estimation of worst-case remaining execution cycles $\check{R}_i$ will need the current number of iterations for each of the outer loops which were performed so far (in order to be able to compute the worst-case remaining iteration counts based on the respective maximal iteration counts).

Our scheme for computing remaining worst-case execution cycles is much simpler: It starts with an estimation of worst-case execution cycles for the task as a whole which is obtained by static analysis. In order to keep track of the remaining worst-case execution cycles $\check{R}_i$, the processor has to provide a cycle counter $\check{C}_{CLK}$ which is incremented during program execution. For each task $\tau_i$, the operating system then maintains a task-specific cycle counter $\check{C}_{act,i}$ derived from $\check{C}_{CLK}$. Based on the initial number of worst-case execution cycles for $\tau_i$, the number of cycles used for $\tau_i$, and the number of cycles saved in $\tau_i$ (compared to the worst-case estimation), the remaining worst-case execution cycles $\check{R}_i$ are computed. The number of cycles saved in $\tau_i$ is updated at frequency scaling points by additional code as will be described below.

In more detail, the counter $\check{C}_{act,i}$ is reset to 0 whenever $\check{R}_i$ is updated (see also Section III-E). This happens in the following three cases: 1.) A task $\tau_i$ is activated – here, $\check{R}_i$ is initialized to $\check{C}_i$. 2.) A task $\tau_i$ is preempted. Assume that $\tau_i$ has already executed $\check{C}_{act,i}$ cycles since the last update, then the remaining cycles are updated with $\check{R}_i \leftarrow \check{R}_i - \check{C}_{act,i}$. 3.) Information on saved cycles is evaluated during task execution. Here, the remaining execution cycles $\check{R}_i$ are updated at Frequency Scaling Points (FSP).

*a) R-type Frequency Scaling Points:* We call the Frequency Scaling Points (FSPs) mentioned above R-type FSPs, because they give relative information about saved cycles in the control flow. Consider that the executed task traverses the edge $b_{if,1} \rightarrow b_2$ in Figure 1. The block $b_1$ would be omitted in this case and 100 cycles are saved. Such relative information is independent of the number of (potentially nested) loop iterations performed so far. Whenever a task $\tau_i$ reaches a R-type FSP which saves $\check{C}^R$ cycles, it updates the remaining cycles $\check{R}_i$:

$$\check{R}_i \leftarrow \check{R}_i - \check{C}_{act,i} - \check{C}^R \quad \text{if R-type FSP}$$

It is easy to see that the scheme based on cycle counters and relative information can be implemented with low overhead for computing the remaining worst-case execution cycles. Another key advantage consists in the fact that it is *suitable for shared code* in libraries used by different tasks. In this case absolute information about remaining worst-case execution cycles can not be written into the shared code, because the information differs depending on the context in which the shared code is used. Thus, the annotation scheme from [10] is not applicable. In our scheme shared code contains only relative information about saved cycles and the remaining worst-case execution cycles are computed based on this as described above.

*b) A-type Frequency Scaling Points:* For completeness, we mention a further optimization for special cases within non-shared code. For these cases we introduce A-type FSPs which keep absolute information about remaining worst-case execution cycles. Consider the edge $b_{wh} \rightarrow b_{if,2}$ in Figure 1. Here, the number of remaining cycles can directly be determined (75 cycles). An A-type FSP can only be used if the corresponding edge is not located inside a loop or inside shared code. The absolute number of remaining worst-case execution cycles is directly updated by

$$\check{R}_i \leftarrow \check{C}^A \quad \text{if A-type FSP.}$$

A-type FSPs make sense when $R$-type FSPs do not provide perfect knowledge about saved cycles (e.g., if not all branches in the control flow are annotated with $R$-type FSPs or if worst-case execution time computation is imprecise due to advanced features of processors like caches and pipelines).[2]

[2] For shared code, A-type FSPs can be transformed into R-type FSPs where $\check{C}^R$ is computed depending on the number of saved cycles.

### C. Inter-Task Slack-time Elimination

In the previous Sect. III-A and III-B we determined the frequency $f_\alpha$ to eliminate idle times and explained how the remaining execution cycles $\check{R}_i$ of a task can be updated during run time. Now we will look into the question of how to compute frequencies based on worst-case remaining cycles. To be able to use frequency scaling in a multi-task environment in combination with inter-task frequency scaling we do not use the old estimation for the intra-task scaling factor as a basis for computing the new frequency, but we use a 'remaining time' $t_{remain,i}$ which depends on inter-task slack-time elimination and on effects of intra-task slack-time elimination with discrete frequencies. Here, the lowest possible frequency to run a single task $\tau_i$ with $\check{R}_i$ remaining cycles and a remaining time $t_{remain,i}$ is $f = \check{R}_i / t_{remain,i} = \check{R}_i/(e_i - t)$ with an ending time $e_i$ and the current time $t$.

In order to determine $e_i$ and to consider inter-task slack-time we use the same principle as the EDF based frequency scaling algorithm given in [8]. Whenever the actual completion time of a task is less than its worst-case completion time $e_i$, the additional slack time is implicitly passed to the subsequent task. The (absolute) worst-case completion time $e_i$ can be calculated incrementally whenever a task $\tau_i$ starts, preempts (another task) or resumes at current time $t$. The computation of $e_i$ makes use of execution times $C_i$ which are reserved for tasks $\tau_i$. $C_i$ results from the worst-case execution cycles $\check{C}_i$ by $C_i = \check{C}_i \cdot f_\alpha^{-1}$ where $f_\alpha$ is the frequency determined as described in Sect. III-A. (If all tasks run with frequency $f_\alpha$ and use their worst-case execution cycles, then the processor utilization $U$ exactly equals 1 and EDF scheduling is able to find a feasible schedule.) The absolute worst-case completion times $e_i$ are computed as follows:

$$e_i = \begin{cases} t + C_i & \text{if } \tau_i \text{ preempts task } \tau_j \\ e_i + e_k - t_i^p & \text{if } \tau_i \text{ was preempted at } t_i^p \\ & \text{and resumes after task } \tau_k \\ e_k + C_i & \text{if } \tau_i \text{ starts after } \tau_k \\ & \text{and } d_i \geq d_k \wedge t < e_k \\ t + C_i & \text{otherwise} \end{cases} \quad (1)$$

In cases 2 and 3 this scheme ensures that a possible slack is used by the following task. The difference of $e_i - t$ gives the available time for task $\tau_i$ and thus, the lowest possible frequency is $f = \check{R}_i/(e_i - t)$. The absolute worst-case completion time $e_i$ is used during intra-task slack-time elimination to further decrease the frequency whenever it detects saved cycles at frequency scaling points.

The correctness of the overall approach follows from the correctness of [8] together with the observation that intra-task frequency scaling never increases the computation time of a task $\tau_i$.

### D. Split Frequency Rescaling

In [15], Ishihara and Yasuura considered the problem of minimizing energy consumption for variable voltage processors. They proved that the *two voltages which minimize energy consumption under a time constraint are immediate neighbors to the $v_{ideal}$* – the optimal voltage for a continuous voltage processor. A similar scheme is applied for discrete frequencies in [16], so that a task with a deadline constraint is firstly executed with a lower frequency and later switched to a higher frequency to meet its deadline for the worst-case. In contrast to [16] we also perform splitting at each frequency scaling point. More precisely, we split the time needed to execute the remaining worst-case execution cycles $\check{R} = \check{R}_a + \check{R}_b$ which should be executed with frequency $f$ into two time intervals executed at the largest frequency $f_a \leq f$ and the smallest frequency $f_b \geq f$. We conclude under the constraints $\check{R} = \check{R}_a + \check{R}_b$ and $\check{R} \cdot f^{-1} \geq \check{R}_a \cdot f_a^{-1} + \check{R}_b \cdot f_b^{-1}$ that we have to switch to the higher frequency $f_b$ when the number of remaining execution cycles is

$$\check{R}_b = \left\lceil \check{R} \cdot \frac{f^{-1} - f_a^{-1}}{f_b^{-1} - f_a^{-1}} \right\rceil .$$

Therefore, the tasks starts with $f_a$ and, if there are only $\check{R}_b$ cycles left, the scheduler switches to the higher frequency $f_b$. For a real processor, this frequency switch can be triggered with a previously set timer.

Note that split frequency rescaling aims to eliminate slack times due to discrete frequencies and – in the ideal case – no inter-task slack should occur. However, this goal can not always be achieved, as the computed ideal frequency can be lower than the lowest available frequency $f_1$ and therefore, a task instance would still produce slack after its completion. Another reason to consider inter-task slack would be the application of this approach for complex architectures, e.g. with caches. Firstly, static analysis tools tend to overestimate worst-case execution times. Secondly, a task can not detect all saved cycles as their number depends on the preceding cache accesses.

### E. Integration into Operating System

There are two procedures which implement our overall approach based on the concepts for idle, intra- and inter-task slack-time elimination. The first procedure is called whenever a new task arrives or is removed from the task set: it verifies if the new task set is feasible ($U \leq 1$) and calculates the frequency $f_\alpha$ and the reserved computation time $C_i = \check{C}_i \cdot f_\alpha^{-1}$ for each task $\tau_i$. The second procedure is given in Algorithm 1 which combines the concepts of inter- and intra-task slack time elimination whenever there is a context switch or program execution reaches a FSP. Essentially, it updates the values for the remaining cycles $\check{R}_i$ and the absolute finishing time $e_i$ of each task $\tau_i$. Please note that for each context switch and each execution at an intra-task frequency scaling point the needed computations can be performed in constant time.

### F. Overhead Considerations

In order to include overhead due to intra-task frequency scaling points and inter-task context switches a fixed number of cycles can be added to the worst-case execution cycles $\check{C}_i$ of a task $\tau_i$ as discussed in the following.

**Algorithm 1** Called on inter- or intra-task scheduling

```
$calculate\_frequency(\tau_i)$
  **if** intra-task scaling of $\tau_i$ **then**
    // task $\tau_i$ saved some cycles at a FSP
    **if** absolute FSP **then**
      $\check{R}_i \leftarrow \check{C}^A$
    **else**
      $\check{R}_i \leftarrow \check{R}_i - \check{C}_{act,i} - \check{C}^R$
    **end if**
  **else if** inter-task scaling of $\tau_i$ **then**
    **if** $\tau_i$ preempts $\tau_j$ **then**
      $e_i \leftarrow t + C_i$
      $\check{R}_i \leftarrow \check{C}_i$
      $\check{R}_j \leftarrow \check{R}_j - \check{C}_{act,j}$ // $\check{C}_{act,j}$ since last switch
    **else if** $r_i$ resumes after some task $\tau_k$ **then**
      $e_i \leftarrow e_i + e_k - t_i^p$ // $\tau_i$ was preempted at $t_i^p$
    **else**
      // $\tau_i$ starts execution after some task $\tau_k$ has finished
      **if** $d_i \geq d_k$ and $t < e_k$ **then**
        $e_i \leftarrow e_k + C_i$
      **else**
        $e_i \leftarrow t + C_i$
      **end if**
      $\check{R}_i \leftarrow \check{C}_i$
    **end if**
  **end if**
  $\check{C}_{act,i} \leftarrow 0$
  $f \leftarrow \frac{\check{R}_i}{e_i - t}$
  determine $\check{R}_{i,b}$ with next lower frequency $f_a$ and higher frequency
  $f_b$ and set timer which switches to frequency $f_b$ when $\check{R}_{i,b}$ cycles
  are left
  $set\_frequency(f_a)$
```

For *inter-task overhead* we consider the worst-case number of cycles $\check{C}_{\text{CS}}$ for a context switch. Each context switch is always connected with an activation or termination of a task, i.e., the overhead can be estimated by $2 \cdot \check{C}_{\text{CS}}$ per task instance. This consideration includes a switch between tasks, a switch from and to a (possibly virtual) idle task, and preemption of a task. Hence, an upper bound for the additional system utilization $U_{\text{CS}}$ can be determined by $U_{\text{CS}} = \frac{2\check{C}_{\text{CS}}}{f_{\text{CS}}} \sum_{i=1}^{n} \frac{1}{T_i}$, assuming that scheduler routines are executed with a frequency of at least $f_{\text{CS}}$.

Of course, tools for WCET estimation are also able to analyze code with intra-task instrumentation and thus, additional cycles for *intra-task overhead* due to frequency scaling points can be determined. However, only conditional branches in the CFG where the overhead for processing a FSP is smaller than the number of saved cycles will be instrumented. This will lead to an incomplete instrumentation in practical applications.

## IV. Experiments

For our experiments we implemented a simulation framework which determines the energy efficiency with different task sets and hardware configurations. In order to evaluate the effectiveness of our approach, we implemented various well-known DVS schedulers, e.g., LaEDF [4] and OLDVS [8], and compared the results to our scheduler implementation ItcaEDF (*Intra-Task Characteristics Aware EDF*). Before we discuss the results in Section IV-C we will give a more detailed description of our simulation framework and system setup.

### A. Simulation Framework

Our experiments were performed with a compiler and simulation framework written in C++. As our approach utilizes intra-task slack due to unused execution paths during program execution, it was not only necessary to model task execution times but also their control flow. To achieve this, task behavior is described in a language based on ANSI C in order to define task behavior including control flow, task semantics and temporal characteristics. Syntax extensions make to possible to specify execution times for statements and to annotate code with *flow facts* like upper bounds for loops. These extensions are used for both integrated path-based WCET analysis and simulation. Finally, in our framework we implement frequency scaling points with a code instrumentation scheme, i.e., after path-based WCET analysis we instrument the task with additional statements giving hints about the number of saved cycles $\check{C}^R$. The number of saved cycles at a FSP is constant for conditional branches in the CFG. For loops, it is dynamically computed depending on the worst-case numbers of iterations and the actual numbers of iterations.

### B. System and Task Setup

In our simulations we use a processor model with four different operating configurations $S = \{(250 \text{ kHz}, 2\,\text{V}), (500 \text{ kHz}, 3\,\text{V}), (750\,\text{kHz}, 4\,\text{V}), (1\,\text{MHz}, 5\,\text{V})\}$. The energy consumption is computed assuming each cycle needs an energy amount proportional to the square of the operating voltage. As we are only interested in the relative energy consumption of the DVS schedulers, we assume that idling the processor consumes no energy as discussed in [4]. In the following we will provide details on how we generated task sets in our implementation.

Firstly, we choose an overall worst-case utilization factor $U \leq 1$. This utilization factor is randomly split under the given number of tasks, so that $U = \sum_{\tau_i \in \Gamma} U_i$ and $\forall \tau_i, \tau_j : \frac{1}{l} \cdot U_j \leq U_i \leq l \cdot U_j$ for a given $l \in \mathbb{R}$. We choose $l = 2$ to avoid exceeding differences in task characteristics. Subsequently, the system determines a period $T_i \in [100\,\text{ms}, 1000\,\text{ms}]$ and computes the worst-case execution time according to $C_i = U_i \cdot T_i$. Finally, this worst-case execution time is realized by a nested loop:

```
**for** $i = 1, .., i_{\text{max}} = b_{\text{outer}}$ **do**
  // randomly determine $j_{\text{max}} \in [b_{\text{inner,min}}, b_{\text{inner,max}}]$
  // automatically inserted intra-task frequency scaling point
  **for** $j = 1, .., j_{\text{max}} \leq b_{\text{inner}}$ **do**
    // computation with fixed number of cycles $\check{C}_{i,\text{loop}}$
  **end for**
**end for**
```

The real worst-case numbers of cycles $\check{C}_{i,\text{loop}}$ needed in the nested loop and the final overall worst-case utilization factor were computed so that $C_i \cdot f_m = b_{\text{inner}} \cdot b_{\text{outer}} \cdot \check{C}_{i,\text{loop}}$.

In this paper we specifically set $b_{\text{outer}} = 5$ and $b_{\text{inner}} = 10$ and therefore, the inner loop including the FSP is executed five times. The values of $b_{\text{inner,min}}$ and $b_{\text{inner,max}}$ were varied to realize different ranges for the actual execution time (AET) to worst-case execution time (WCET) ratio. For example, values of $b_{\text{inner,min}} = 4$ and $b_{\text{inner,max}} = 8$ can be used to force an average fraction of $0.6$ for the actual workload of a task.

### *C. Results*

The simulation results are given in Figure 2 for different task scenarios. Each diagram shows a histogram for the energy efficiency of a specific scheduler. On the x-axis we have the worst-case utilization factor of the corresponding task set and on the y-axis the energy consumption for the different methods as a fraction of the energy consumption for an EDF based execution with the highest possible frequency $f_m$. In the experiments we varied the number of executed tasks $n$ and the 'actual workload'. The 'actual workload' denotes the fraction $c$ of the actual execution time (AET) to the worst-case execution time (WCET) of a task as described above. The diagrams in this paper refer to scheduling with either $n = 2$ or $n = 8$ tasks and an actual workload either in the interval $[0.8, 1.0]$ or $[0.4, 0.8]$.

We separated the results for the different schedulers into two blocks, respectively. Before we discuss the results we will give a short overview of the compared energy-efficient scheduling methods. The first block consists of real-time DVS schedulers known from literature:

- **StaticEDF:** StaticEDF uses a constant frequency depending on the worst-case utilization $U$ of the task set.
- **OrigIntra**: Original Intra-Task DVS which uses fixed ratios for frequency rescaling [10].
- **OLDVS**: Slack passing scheme which gives unused computation to the subsequent task [8].
- **OLDVS*:** Improved OLDVS variant for discrete frequency processors which splits execution time in two parts executed with the next lower and next higher discrete frequency [16].
- **LaEDF**: Speculative Look-ahead EDF scheduler which scales to the lowest possible frequency by deferring as much work as possible after the next deadline [4].

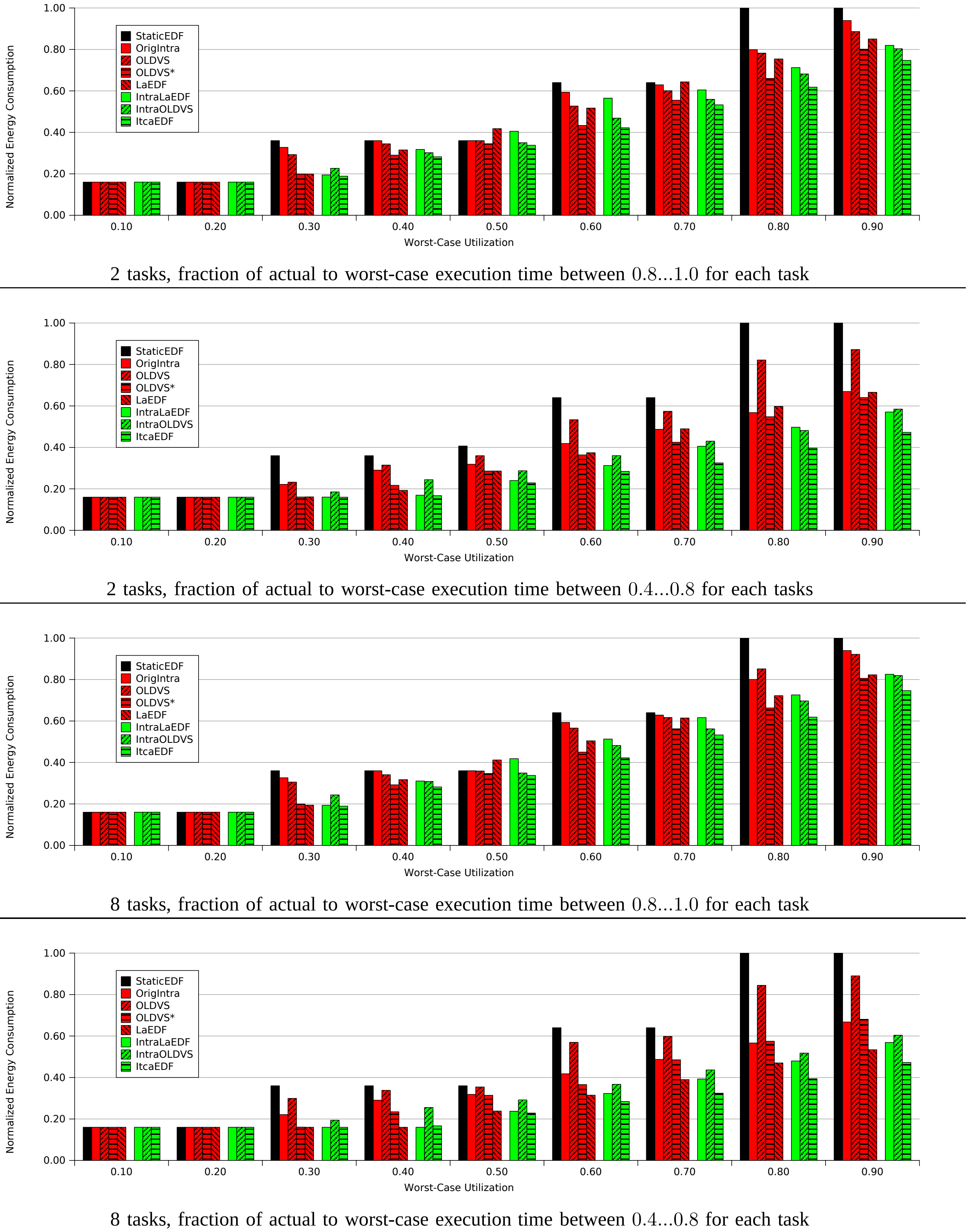
The second block consists of inter-task schedulers which include intra-task voltage scaling according to the scheme presented in this paper. Here, the last method *ItcaEDF* is our proposed approach which outperforms all other DVS schedulers:

- **IntraLaEDF**: Variant of LaEDF combined with our intra-task approach. Rescaling with deferring work after the next deadline is done at each frequency scaling point and at each context switch.
- **IntraOLDVS**: OLDVS combined with our intra-task approach.
- **ItcaEDF (Intra-Task Characteristics Aware EDF)**: The proposed on-line algorithm which integrates our intra-task approach and split frequency rescaling (see Sect. III-D).

In all configurations the minimal energy consumption depends on the lowest available frequency. Therefore for utilization factors smaller than the ratio $f_1/f_m = 0.25$ all frequency scaling techniques gave the same energy consumption as all tasks already start with the lowest possible frequency. The energy consumption increases with higher worst-case utilization factors. As expected, StaticEDF shows a stepwise increase of energy consumption (according to the discrete frequency distribution), because only worst-case execution times and no occurring slack is considered. (Energy consumptions for $U = 0.10, 0.20$, $U = 0.30, 0.40, 0.50$, $U = 0.60, 0.70$, and $U = 0.80, 0.90$, respectively, are identical.)

Considering the first block of schedulers known from literature, neither OrigIntra nor OLDVS show an overall good performance. OrigIntra performs comparatively well when the actual workload is low (see second and fourth histogram), since it is able to rescale with a high granularity at task-internal frequency scaling points. However, OrigIntra shows high energy consumption when the actual workload is large (i.e., not far away from the worst-case workload, see first and third histogram), as the occurring slack for a single task is too small in order to scale to a lower frequency and slack is not accumulated between tasks (since it is not passed to a subsequent task). OLDVS shows only moderate savings compared to StaticEDF, since frequency scaling with slack passing to the subsequent task is performed only at context switches with a lower granularity. The original OLDVS scheduler is improved by the split frequency scaling of OLDVS* which provides a better support for discrete frequencies on the one hand and on the other hand works more optimistically by selecting the lower of the two neighbor frequencies first. The most optimistic (or in other words most aggressive) method is LaEDF which works especially well when the actual workload is low and the number of tasks is high (see fourth diagram). However, when the workload is too high, the speculative scheme of LaEDF (which defers as much work as possible after the next deadline) seems to be too aggressive and the likelihood that, for example, a preempting task has to choose a higher frequency is increased (see first and third diagram). When the number of tasks is low (see first and second diagram) LaEDF also may suffer from the fact that the number of context switches becomes smaller and there are less opportunities for frequency scalings.

Considering the last blocks, our proposed ItcaEDF approach clearly outperforms the other schedulers in all task setups. For instance, considering a worst-case utilization of $U = 0.8$, a AET/WCET ratio between 0.4 and 0.8, and two tasks (see diagram 2), ItcaEDF improved the normalized energy consumption by 28% compared to OLDVS* and 34% compared to LaEDF. Increasing the number of tasks to 8 (see diagram 4) results in a energy reduction of 16% compared to LaEDF and 31% compared to OLDVS*. Apparently, in our scheduler the deep integration of intra-task code instrumentation into inter-task frequency scaling really pays off. The intra-task frequency scaling points both provide timing information to reflect the actual work load of the system more accurately and give a



2 tasks, fraction of actual to worst-case execution time between $0.8 \ldots 1.0$ for each task

2 tasks, fraction of actual to worst-case execution time between $0.4 \ldots 0.8$ for each tasks

8 tasks, fraction of actual to worst-case execution time between $0.8 \ldots 1.0$ for each task

8 tasks, fraction of actual to worst-case execution time between $0.4 \ldots 0.8$ for each task

Fig. 2. Normalized energy consumption under various task characteristics for 2 or 8 tasks and different fractions of utilization

higher granularity for efficient scheduling with DVS. ItcaEDF avoids working too aggressively (as LaEDF in some cases), since the only case of an optimistic frequency selection occurs when a not existing continuous frequency is approximated by two discrete frequencies and the lower one is selected first. However this decision is confirmed by the observation that the need for using the second (higher) frequency is often canceled in the future, when the actual workload is low and additional slack occurs after the frequency splitting.

For completeness, we also considered schedulers where our intra-task code instrumentation scheme is integrated into OLDVS directly (without frequency splitting) and into LaEDF (the corresponding bars in the histograms are labeled IntraOLDVS and IntraLaEDF, respectively). As expected, IntraOLDVS (in comparison to OLDVS) profits from the detection of reduced remaining worst-case execution cycles during task run time, but it is outperformed by ItcaEDF. Somewhat surprisingly, the results for IntraLaEDF show that it is not always a good idea to combine an inter-task scheduler with intra-task code instrumentation: LaEDF can hardly benefit from our code instrumentation approach and sometimes IntraLaEDF displays an even higher energy consumption than LaEDF. Obviously, increasing the aggressiveness of the speculative algorithm by exploiting intra-task characteristics has a negative impact on energy-efficiency.

## V. Conclusions

In this paper, we proposed an energy-efficient real-time scheduler which incorporates cycle-based intra-task and time-based inter-task frequency scaling for the realistic scenario of processors with discrete frequencies and dynamic task sets. A multi-level approach considers idle, intra- and inter-task slack times and scales the processor speed to the lowest possible frequency on context switches and within task execution. We also presented a novel technique to keep track of remaining worst-case execution cycles for intra-task frequency scaling which is based on cycle counters of the hardware and which is suitable for shared code. To evaluate the performance of the on-line scheduler we integrated the algorithms into a compiler and simulation framework. Our experimental results showed that our approach is able to reduce energy consumption of state-of-the-art inter-task DVS schedulers by over 30%.

## Acknowledgement

This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103.

## References

[1] T. D. Burd and R. W. Brodersen, "Energy efficient CMOS microprocessor design," in *HICSS '95: Proceedings of the 28th Hawaii International Conference on System Sciences*. Washington, DC, USA: IEEE Computer Society, 1995, p. 288.

[2] K. Govil, E. Chan, and H. Wasserman, "Comparing algorithm for dynamic speed-setting of a low-power CPU," in *MobiCom '95: Proceedings of the 1st annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 1995, pp. 13–25.

[3] T. Pering and R. Broderson, "Energy efficient voltage scheduling for real-time operating systems," in *Proceedings of the 4th IEEE Real-Time Technology and Applications Symposium RTAS'98, Work in Progress Session*, Jun. 1998.

[4] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2001, pp. 89–102.

[5] Y. Zhu and F. Mueller, "Feedback EDF scheduling of real-time tasks exploiting dynamic voltage scaling," *Real-Time Syst.*, vol. 31, pp. 33–63, 2005.

[6] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez, "Power-aware scheduling for periodic real-time tasks," *IEEE Trans. Comput.*, vol. 53, no. 5, pp. 584–600, 2004.

[7] C. Scordino and G. Lipari, "A resource reservation algorithm for power-aware scheduling of periodic and aperiodic real-time tasks," *IEEE Trans. Comput.*, vol. 55, no. 12, pp. 1509–1522, 2006.

[8] C.-H. Lee and K. G. Shin, "On-line dynamic voltage scaling for hard real-time systems using the EDF algorithm," in *RTSS '04: Proceedings of the 25th IEEE International Real-Time Systems Symposium*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 319–327.

[9] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[10] D. Shin, J. Kim, and S. Lee, "Intra-task voltage scheduling for low-energy hard real-time applications," *Design & Test of Computers, IEEE*, vol. 18, pp. 20–30, 2001.

[11] D. Shin and J. Kim, "Optimizing intra-task voltage scheduling using data flow analysis," in *ASP-DAC '05: Proceedings of the 2005 conference on Asia South Pacific design automation*. New York, NY, USA: ACM, 2005, pp. 703–708.

[12] ——, "Optimizing intratask voltage scheduling using profile and data-flow information," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, pp. 369–385, 2007.

[13] J. Seo, T. Kim, and N. D. Dutt, "Optimal integration of inter-task and intra-task dynamic voltage scaling techniques for hard real-time applications," in *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 450–455.

[14] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *Solid-State Circuits, IEEE Journal of*, vol. 27, pp. 473–484, 1992.

[15] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *ISLPED '98: Proceedings of the 1998 international symposium on Low power electronics and design*. New York, NY, USA: ACM, 1998, pp. 197–202.

[16] M.-S. Gong, Y. R. Seong, and C.-H. Lee, "On-line dynamic voltage scaling on processor with discrete frequency and voltage levels," in *ICCIT '07: Proceedings of the 2007 International Conference on Convergence Information Technology*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1824–1831.

# 3$^{rd}$ International Workshop on Secure Information Systems

THE SIS workshop is envisioned as a forum to promote the exchange of ideas and results addressing complex security issues that arise in modern information systems. We aim at bringing together a community of security researchers and practitioners working in such divers areas as networking security, antivirus protection, intrusion detection, cryptography, security protocols, and others. We would like to promote an integrated view at the security of information systems.

As information systems evolve, becoming more complex and ubiquitous, issues relating to security, privacy and dependability become more critical. At the same time, the global and distributed character of modern computing - typically involving interconnected heterogeneous execution environments - introduces many new and challenging engineering and scientific problems. Providing protection against increasingly sophisticated attacks requires strengthening the interaction between different security communities, e.g. antivirus and networking. New technologies lead to the emergence of new threats and attack strategies, involving smart mobile devices, peer-to-peer networks, instant messaging, VoIP, mesh networks or even networked consumer devices, such as house appliances or cars. Furthermore, the increasing openness of the communications infrastructure results in novel threats and can jeopardize critical enterprise and public infrastructure, such as industrial automation and process control systems. Not only it is estimated that half of all Web applications and Internet storefronts still contain some security vulnerabilities, but secure commerce applications are also exposed to increasingly elaborate attacks, including spyware, phishing and other social engineering methods.

In order to develop a secure system, security has to be considered in all phases of the lifecycle and adequately addressed in all layers of the system. At the same time, good engineering has to take into account both scientific and economic aspects of every solution: the cost of security has to be carefully measured against its benefits - in particular the expected cost of mitigated risks. Most companies and individuals treat security measures in information system as a necessary, but often uncomfortable, overhead. The increasing penetration of computing in all domains of everyday life means that security of critical business systems is often managed and maintained by personnel who are not knowledgeable in the field. This highlights the importance of usability and ease of configuration of security mechanism and protocols.

Covered topics include (but are not limited to):

- Access control
- Adaptive security
- Cryptography
- Copyright protection
- Cyberforensics
- Honeypots
- Intrusion detection
- Network security
- Privacy
- Secure commerce
- Security exploits
- Security policies
- Security protocols
- Security services
- Security evaluation and prediction
- Software protection
- Trusted computing
- Threat modeling
- Usability and security
- Viruses and worms
- Zero-configuration security mechanisms

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 775 – 782

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Context-Risk-Aware Access Control Model for Ubiquitous Environments

Ali Ahmed
University of Manchester, School of Computer Science, Oxford Road, Manchester, M13 9PL, UK
Email: ahmeda@cs.man.ac.uk

Ning Zhang
University of Manchester, School of Computer Science, Oxford Road, Manchester, M13 9PL, UK
Email: nzhang@cs.man.ac.uk

***Abstract*—This paper reports our ongoing work to design a *C*ontext-*R*isk-*A*ware *A*ccess *C*ontrol (*CRAAC)* model for Ubiquitous Computing (UbiComp) environments. *CRAAC* is designed to augment flexibility and generality over the current solutions. Risk assessment and authorisation level of assurance play a key role in *CRAAC*. Through risk assessment, resources are classified into groups according to their sensitivity levels and potential impacts should any unauthorised access occurs. The identified risks are mapped onto their required assurance levels, called Object Level of Assurance (*OLoA*). Upon receiving an object access request, the requester's run-time contextual information is assessed to establish a Requester's Level of Assurance (*RLoA*) denoting the level of confidence in identifying that requester. The access request is granted iif *RLoA* ≥ *OLoA*. This paper describes the motivation for, and the design of, the *CRAAC* model, and reports a case study to further illustrate the model.**

## I. Introduction

UBICOMP, sometimes referred to as pervasive computing, envisages a new computational environment in which heterogeneous devices with varying levels of capabilities and sensitivities interact seamlessly to provide smart services. By gathering information about surroundings (contextual data), the environment adaptively and non-intrusively provides context-aware services to users [1]. A context, defined as " *any information that can be used to characterise the situation of an entity* " [2], could relate to users (e.g. a ccess location ) or to systems (e.g. network channel security level).

Context is dynamic, and its values may change from one session to another, and even during the same session. While UbiComp adapts its services to the surrounding context, the security services in the underlying environment should also be adaptive to the relevant context. In Access Control (AC) for example, we emphasis that an AC solution for UbiComp environments should be context-aware; it should react not only to value changes of individual contextual attributes, but also to the composite effect as caused by multiple contextual attributes value changes. It is inappropriate for an AC solution to take multiple contextual attributes directly as additional AC constraints while disregarding their composite effect on the risk level of unauthorised access in the underlying system. Furthermore, users' mobility further exacerbates the AC challenges in such environment. Mobile users with known/unknown devices move in/out of the environment without the protection of infrastructure-based firewalls exposes the environment to more security threats and attacks. An effective UbiComp AC solution should take into account the level of confidence in the entity trying to gain access to sensitive resources as well as the confidence level in the provided contextual information.

To realise this vision of context-aware AC, we have designed the *CRAAC* model, which uses the notion of risks and risk-linked levels of assurance (*LoA*) to govern the AC decisions. Through risk assessment, resources/services are classified into different groups each with a distinctive *OLoA*. In fact, the *OLoA* of a given resource object is determined based on its sensitivity level and the potential harm or impact should any unauthorised access to that object occurs. When an object access request is received, the requester's run-time contextual information is assessed to establish an *RLoA* denoting the confidence level in that requester. The access request is granted iif *RLoA* ≥ *OLoA*.

This idea of using a risk linked *LoA* is inspired by the OMB/NIST e-Authentication Guidelines in [3], [4]. Similar to the OMB/NIST approach, *CRAAC* uses risk assessment to identify the risks for resource objects, and maps the identified risks to appropriate assurance levels (*OLoA)* for that object group. Our work, however, features the following distinct characteristic over the OMB/NIST work. The OMB/NIST work addresses the issue of authentication in the context of electronic transactions in a static environment, whereas our work focuses on context-aware AC in a dynamic UbiComp environment. As a result of this fundamental difference, our work differs from the OMB/NIST effort in the following ways. Firstly, the OMB/NIST guidance only considers issues related to users identification via the use of electronic credentials (e-credentials) that are largely static, whereas we handle a broader range of AC attributes, not only static e-credentials but also dynamic contextual attributes. Secondly, unlike the OMB/NIST work that only considers risk impact as caused by a single attribute (i.e. e-authentication credentials), we consider risk impacts by multiple attributes, as well as their composite effect on the authorisation assurance level. In addition, unlike the OMB approach by

Would like to thank the Faculty of Computers and Information, Cairo University, for its financial support.

which appropriate authentication technologies are chosen and implemented prior run-time to ensure that the underlying system achieves the required level of authentication assurance, *CRAAC* derives an *RLoA* for each requester at run-time based on their real-time dynamic contextual information, then compares this *RLoA* against *OLoA,* an AC threshold for the requested object, to make an AC decision.

The rest of this paper is structured as follows: section II gives an overview of the related work in context-aware AC. Section III describes the *CRAAC* model in detail. A case study is given in section IV. Finally, section V concludes the paper and outlines the future work.

## II. Related Work

The main objective of an AC system is to restrict the actions a legitimate user can perform on a given resource object [5]. Role-Based Access Control (RBAC) [6] is a powerful model to specify and enforce organisational policies in a way that seamlessly maps to an enterprise structure [7]. Instead of assigning access rights to users directly, RBAC assigns access rights to roles that users can have as part of their organisational responsibilities. RBAC is considered as a policy natural authorisation approach particularly suited to large-scaled distributed environments [8]. However, the major weakness of the RBAC model is that it can not capture any security relevant information from its environment due to the subject-centric nature of its roles [8]. As a result, it can not enforce context-aware security policies, and therefore it is not adequate for UbiComp environments. To overcome this weakness, there have been proposals, will be discussed later on, to extend the basic RBAC model to equip it with the context-awareness capability. Those proposals use contextual information directly as additional constraints to govern the AC decision, as depicted by Fig. 1.

One of the earliest proposals is the Generalized Role-Based Access Control (GRBAC) model [9]. It introduced the concept of environment roles to capture contextual information from the underlying access environment. GRBAC, However, requires the use of complex system architecture to support the extended roles [10].

Another notable proposal is the Temporal RBAC (TRBAC) model [11] which extends the traditional RBAC model by introducing a temporal constraint into the AC specification to provide a mechanism to enforce time-dependent AC policies [12]. A subsequent proposal, the Generalized Temporal RBAC (GTRBAC) model [13], further extends the TRBAC model by introducing the notion of an activated role. More precisely, GTRBAC differentiates enabled roles, which subjects can activate, from active roles, which are being activated by at least one subject, for a more fine-grained AC.

The Spatial RBAC (SRBAC) model in [14] introduces a location-dependent constraint. A location space is divided into multiple *zones* , and an access permission is granted if the role condition is satisfied and the user is within the specified *zone* . The SRBAC model, as observed by [15], suffers from a lack of a semantic meaning of the position information, and it does not support the use of geometrically bounded roles.

The Dynamic Role Based Access Control (DRBAC) model [16], specifically designed for AC adaptation in UbiComp environments, is an interesting piece of work. Any change in an access context will be captured by an '*agent*' that will, in turn, trigger an '*event'* to cause a transition between the current role/permissions set to a new role/permissions set. DRBAC is considered as a pioneering effort in achieving context-aware authorisation in UbiComp environments. However, as noted by the authors themselves, implementing DRBAC can significantly increase the complexity of the applications concerned. This is particularly troublesome for the resource-restricted devices typically seen in UbiComp environments.

Young-Gab *et al* in [17] proposed a context-aware AC model which considers location, time, and system resources as AC constraints. The role is activated only if all the constraints are satisfied. The model has failed to consider the potential composite effects of, or the correlations between, these context attributes. In a similar approach, the model in [12] divides the location information in a levelled manner. The work formalised the model and conducted a case study, but again, it only focuses on temporal and spatial context attributes.

The work by G. Motta in [18] focuses on preserving patients' privacy and protecting the confidentiality of the patients' data in smart hospitals. The proposed contextual RBAC model classifies the patients' records based on their sensitivity levels, and an AC decision is made based upon the sensitivity level of the data being requested. The work, however, does not show how to adjust AC decisions in adaptation to the requesters' dynamic changes of the contextual information.

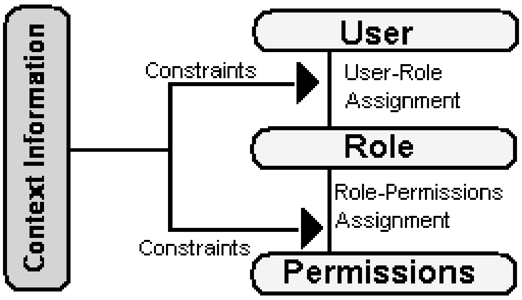


Fig 1. Existing Context-aware RBAC Approaches

A recent published work [19] has tried to address the need for evaluating the effect of multiple contextual attributes on an authorisation decision coherently. The model introduces the notion of risk-aware AC. The context information is used as the input to a risk assessment process to compute a risk value that is then fed into the authorisation decision engine. However, the scope of the risk assessment is quite broad covering confidentiality, integrity and authentication, so the delay incurred in the risk value calculation may be quite large, which may adversely affect the performance of the underlying AC system. Whether this delay would decrease the system ability to promptly adapt its decisions to context changes is yet to be investigated.

From the above discussions, it is clear that more investigation is needed for designing an AC model that could accommodate multiple contextual attributes in a generic and a coherent manner, and adapts its decisions to the dynamic changes of context, while, at the same time, keeping the costs down. The next section describes the design of such a model.

## III. *CRAAC* Model

The *CRAAC* model aims at achieving *context-aware adaptation* (i.e. requirement 1 — capable of capturing, and adapting its decisions to the surrounding information for fine-grained AC), *flexibility* (requirement 2 - flexible enough to accommodate different contextual attributes and should not tie itself to a particular application domain), *extensibility* (requirement 3 - extensible to allow easy addition of new, and removal of obsolete, contextual attributes and any alterations imposed to the architecture as caused by such contextual attribute changes should be refrained within the context management part of the AC system), and *low performance costs* (requirement 4 — performance costs incurred in achieving context-awareness should be kept as low as possible).

### A. Methodology Overview

The architecture of a context-aware AC system should include two major functional blocks, one for the *Context Management (CM)* and the other for AC. *CM* encompasses components for context acquisition, interpretation and representation while *AC* is responsible for making authorisation decisions. *CRAAC* is built on the basic RBAC model. While satisfying the requirements outlined above, the model has the two functional blocks (CM and AC) loosely coupled. That is, any change made to the attributes set managed by the CM block should not lead to changes in AC algorithms and policy representations managed by the AC block, and vice versa. One way to facilitate this loose coupling is to use a generic attribute that, on one hand, can capture the impact on AC as caused by context changes (in CM block), and, on the other hand, to feed that impact into the AC block as an additional AC constraint. In addition, it is desirable to link that attribute value to the resources sensitivity levels. Based upon these considerations, we introduce the notion of authorisation Level of Assurance (*LoA*) and use it as that generic attribute.

In details, resources/services are classified into object groups each with a distinctive *OLoA*. The determination of *OLoA* of an object group can be done via risk assessment for that group. The assessment identifies the risks, assesses their potential impacts, and maps the identified risks to an appropriate assurance level, i.e. *OLoA*. When an object access request is received, the decision engine will compare the *RLoA,* derived based on the requester's contextual information, against the *OLoA* of that object. The request is granted iif $RLoA \geq OLoA$. In fact, the *OLoA* of an object is the minimum authorisation requirement a user has to satisfy to gain access to that object. The more sensitive the object is, and/or the higher the potential impact, the higher the *OLoA*. Consequently, the higher the *RLoA* a requester would have to satisfy before the request can be granted.

One of the challenging tasks for designing this context-aware *LoA* linked AC is how to derive an *RLoA* value for a given access request based upon the requester's real-time contextual information. To achieve this, we need to, firstly, identify a set of contextual attributes that have impacts on the degree of certainty that the access request is from an entity that it claims to be from, secondly, to investigate, analyse, and define the respective assurance levels for these attributes, and thirdly, to devise a method that can derive the *RLoA* value based upon the attributes' LoA. In the remaining part of this section, we are going to address these issues respectively.

### B. Contextual Attributes and their LoA Definitions

There are a number of factors that can increase the risk of unauthorised access, e.g. weak authentication protocol/token, less trustworthy access location, lack of intrusion detection and response systems, unprotected communication channels, … etc. In this paper, the focus is on the authentication token types, the access locations, the channel security, and the ability to respond to intrusion attacks (intrusion response). We name these factors as contextual attributes. At run-time, the risk associated to these attributes, if materialised, may lead to unauthorised information access. In the rest of this section, these attributes will be discussed in details.

#### 1. eToken Attribute

Many factors in an e-authentication process affect the confidence level (i.e. *LoA*) in verifying a claimed identity. These include identity proofing, credentialing, credential management, record keeping, auditing, authentication protocols and token types. The assurance levels of some of these steps are achieved through procedural and process governance, while others may be left to the requesters' decision. For example, a requester may choose to use a particular authentication credential when making an access request. As our focus here is on the derivation of an authentication *LoA* and on linking it to the authorisation decision making, we exclude the procedural factors (i.e. user registration, credential management and storage procedures) from the *LoA* derivation. We rather focus on the types of e-credentials/tokens that are collectively called *eTokens*. Different *eTokens* provide varying degrees of confidence in entity identification and authentication. To quantify that degree of confidence, we introduce the notions of $LoA_{eToken}$.

***Definition 1:*** *$LoA_{eToken}$ refers to the service provider's degree of confidence that an eToken presented by a user is linked to his/her identity.*

The *eToken* types versus their assurance levels have been recommended by NIST [4], as shown in Table I. NIST recognises the token types of hard tokens, soft tokens, one-time password (OTP) device tokens, and user-name/password pairs, and defines four levels of $LoA_{eToken}$.

#### 2. Access Location (ALoc) Attribute

Authentication services are of two main types; one is e-authentication by which a user is identified through the use of an eToken, and the other is physical authentication (p-authentication) by which a user is identified through the use of biometrics, sensors or location based services. *CRAAC* recognises both of these authentication services. This is because, firstly, a combined use of e-authentication and location-based p-authentication not only provides optional services to users, but also offers a more reliable user identification. Secondly, location based services are commonly seen in UbiComp environments, and the access location is an important contextual attribute in such environment. Therefore, in addition to the *eToken* attribute, we introduce another authentication attribute, Access Location (*ALoc*). $LoA_{ALoc}$ in relation to *ALoc* is defined below.

***Definition 2:*** *LoA $_{ALoc}$ refers to the degree of confidence in a claimed a ccess location .*

Depending on the application context, there are various ways to represent the location alternatives [20], [21]. As our As our focus is on the degree of confidence in a claimed location, we use the ‘ *zone* ’ representation method [20], [14] to describe different location alternatives. Table II shows some possible location alternatives versus their likely assurance levels. The table is meant for illustration purpose only, as, unlike the case of *eToken* , *ALoc* attribute does not have any international consensus on their *LoA $_{ALoc}$* .

As outlined above, the two attributes, *eToken* and *ALoc,* both make direct contributions to the overall confidence level in the user identification. For example, as a password token is more vulnerable to guessing attacks than a PKI credential, then using it inside a secure room with a biometric physical authentication facility may be comparable, in terms of authentication assurance level, with a PKI credential used in a public area. To quantify such correlation between the two attributes, we introduce the notion of $LoA_{authN.}$

***Definition 3:*** *LoA $_{authN}$ is the overall confidence level associated with the composite authentication solution consisted of token based e-authentication and location based p-authentication .*

The derivation of $LoA_{authN}$ given $LoA_{eToken}$ and $LoA_{ALoc}$ will be discussed later on.

### 3. *Channel Security (CS) Attribute*

The level of security protection of the channel running between the requester and the service provider may indirectly influence the risk level of unauthorised access. For instance, if a channel is more vulnerable to eavesdropping attacks, some credentials sent over the channel may experience a high risk of being compromised. In addition, the requested data may also experience a high risk of being disclosed to unauthorised entities via channel interceptions. For this reason, we introduce another contextual attribute, Channel Security ( *CS* ) and its *LoA* is defined below.

TABLE I.
TOKEN TYPES VERSUS $\text{LoA}_{\text{eToken}}$ [4]

| Token type | Levels | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| **Hard token** | √ | √ | √ | √ |
| **One-time password token** | √ | √ | √ | |
| **Soft token** | √ | √ | √ | |
| **Password token** | √ | √ | | |

TABLE II.
LOCATION ALTERNATIVES VERSUS $\text{LoA}_{\text{ALoc}}$

| Alternatives | $\text{LoA}_{\text{ALoc}}$ |
|---|---|
| **Zone-0** | **Level 0**; public area which does not have any provision for p-authentication. |
| **Zone-1** | **Level 1**; semi-public area which uses p-authentication to identify a group of users, e.g. through the use of a shared building key. |
| **Zone-2** | **Level 2**; personal area – access to this zone is controlled by the use of a locker key owned by a single user or a sensor based user identification (e.g. RFID). |
| **Zone-3** | **Level 3**; secured personal area – this zone uses some strong form of physical identification method that is less vulnerable to theft or loss than locker keys, e.g. Biometrics (physical) authentication facility. |
| **Zone-4** | **Level 4**; highly secured personal area – this zone may use multiple physical authentication methods. |

***Definition* 4:** *LoA $_{CS}$ refers to the degree of confidence in the channel (linking requesters and service providers) security .*

Similar to the *ALoc* attribute, the *CS* attribute does not have any international consensus on its assurance level definition. Table III describes an exemplar setting of a 5-level $LoA_{CS}$ mimicking the NIST’s *eToken LoA* definition. The exemplar setting is for a demonstration purpose and to be used later on in section V.

### 4. *Intrusion Response Attribute*

The last contextual attribute addressed by *CRAAC* is the Intrusion Response (*IR*) attribute, and its *LoA* is denoted by $LoA_{IR}$.

***Definition 5:*** *$LoA_{IR}$ refers to the degree of confidence in the ability of the underlying system to detect intrusion attacks and the ability to respond to such attacks*.

Sundaram in [22] divides IDS detection mechanisms into; anomaly, and misuse. An IDS may monitor host computers or network activities. The ability to detect intrusions and to respond promptly to these intrusions varies from an IDS to another. An IDS class may have an associated level of confidence as they vary in capabilities. Again there is no international consensus on the $LoA_{IR}$ definition. Table IV is an exemplar $LoA_{IR}$ setting for illustration purposes and to be used later on in the case study.

Once the AC related attributes are identified, and their assurance levels are specified, the next step is to estimate an overall/aggregate *LoA* value for an access requester based upon the assurance levels of such attributes.

## C. *Requesters’ Aggregate LoA Derivations*

### 1. *Relationships among Multiple Attributes*

As mentioned earlier, the confidence level in identifying a user may be influenced by multiple methods or attributes, either directly (eTokens and Access Location) or indirectly (Channel Security and Intrusion Response). To quantify the confidence level as influenced by the combination of a requester’s multiple contextual attributes, we introduce the notion of an overall (aggregate) assurance level for a requester, *RLoA*.

***Definition* 6*:*** *RLoA refers to an overall LoA in identifying a requester based upon the requester's contextual information associated to multiple contextual attributes (eToken, ALoc, CS, and IR).*

The derivation of *RLoA* depends on the types of the attributes used in the access session, the correlation among the attributes, and the used security policy. Formally, given a set of contextual attributes ($A_1, A_2, ..., A_n$) and their associated assurance levels ($LoA_{A_1}, LoA_{A_2}, ..., LoA_{A_n}$), *RLoA* can be expressed using a generic function, *f*, as:

TABLE III.
LOA ASSOCIATED TO CHANNEL SECURITY ATTRIBUTE

| $LoA_{CS}$ | Descriptions |
|---|---|
| **Level 0** | This attribute is disabled, or not used. |
| **Level 1** | Little or no confidence in channel security. |
| **Level 2** | Some confidence in channel security. |
| **Level 3** | High confidence in channel security . |
| **Level 4** | Very high confidence in channel security. |

TABLE IV.
LOA ASSOCIATED TO INTRUSION RESPONSE ATTRIBUTE

| $LoA_{IR}$ | Descriptions |
|---|---|
| **Level 0** | IDS is disabled or no IDS is installed. |
| **Level 1** | Little or no confidence in the installed IDS. |
| **Level 2** | Some confidence in the installed IDS.. |
| **Level 3** | High confidence in the installed IDS. |
| **Level 4** | Very high confidence in the installed IDS. |

$$RLoA = f(LoA_{A_1}, LoA_{A_2}, \dots, LoA_{A_n}) \qquad (1)$$

The function *f* is determined by the relationship among the multiple attributes. We have identified two types of relationships, one is the *elevating* relationship and the other is the *weakest-link* relationship. In the *elevating* relationship, the combined use of two or more contextual attributes may result in the overall confidence level being higher than that provided by any one of the contextual attributes.

*Elevating* security is used by Microsoft in Windows Server 2003 to enable regular users to install applications even if they do not have the required permissions [23]. In our problem context, attributes, *eToken* and *ALoc*, are in an *elevating* relationship, as it is obvious that a combined use of e-authentication and location-based p-authentication will result in a more reliable user identification, thus a higher *LoA*.

In the *weakest-link* relationship, on the other hand, the value of *RLoA* is equal to the lowest attribute *LoA* value in the attributes set. This is in line with the *weakest-link* principle in system security. For example, attributes, {*eToken*, *ALoc*}, *CS* and *IR*, resembles more the *weakest-link* relationship (here *eToken* and *ALoc* are treated as one whole attribute in terms of *LoA*). This is because, even if the underlying authentication procedure is strong (thus difficult to impersonate), and channel security has a high assurance level (thus difficult to intercept useful information), provided that the service provider's system is easy to break into, there will still be a high risk of compromising server end of the identification and authentication procedure, e.g. by directly attacking credential files stored in the system. This implies the overall assurance level should not be higher than the lowest attribute *LoA* involved.

*2. Converting LoA to Ratings*

Later in this section, when we describe the *RLoA* derivation method, the attributes' *LoA* values (e.g. $LoA_{eToken}$, $LoA_{ALoc}$, $LoA_{CS}$, and $LoA_{IR}$) will need to be converted from levels (or ranks), as shown in Tables I-IV, to values in the real interval [0,1] (i.e. ratings or weights). To accomplish this rank-to-rating conversion of *LoA* values, we employ the Rank Order Centroids (*ROCs*) method, a well-known rank-to-rating (or weight) conversion technique. This subsection focuses on describing the *ROCs* method.

*ROCs* is often used in solving an *MCDA* (Multiple Criteria Decision Analysis) problem. It takes a set of attributes ordered by importance (ranks) and converts them into a set of approximated weights (ratings) as sometimes it may not be realistic to determine the precise weights [24]. *ROCs* originally proposed by Barron in [25] with an appealing theoretical rational for its weights [26]. In addition, the weights are derived by a systematic analysis of implicit information in the ranks which would give more accurate outcome [24].

Using the *ROCs* method, the weights are derived from a simplex $w_1 \geq w_2 \geq .... \geq w_n \geq 0$ restricted to:

$$\sum_{i=1}^{n} w_i = 1 \qquad (2)$$

where *n* is the number of attributes (system cardinality). The vertices of the simplex are $e_1 = (1, 0, ..., 0)$, $e_2 = (1/2, 1/2\ 0, ..., 0)$, $e_3 = (1/3, 1/3, 1/3, 0, ..., 0)$, ........ $e_n = (1/n, 1/n, ..., 1/n)$. The coordinates of the centroids (weights) are calculated by averaging the corresponding coordinates of the defining vertices [24]. In general, the weight of the k$^{th}$ most important attribute is calculated as:

$$(\sum_{i=k}^{n} \frac{1}{i})/n \qquad (3)$$

*ROCs* is a light-weight method for the rank-to-rating conversion as *ROC*-based analysis is straightforward and efficacious [24]. Therefore, it is particularly suited to the UbiComp environment. In addition, the weights can be calculated off-line, and uploaded into a rank-to-weight conversion table as shown in Table V to further reduce run-time overheads incurred in the conversion.

Tables, VI and VII, describe the corresponding $LoA_{eToken}$*, and* $LoA_{ALoc}$ values in ratings converted by *ROCs receptively.*

It is worth noting that level$_4$ in both cases of $LoA_{eToken}$ and $LoA_{ALoc}$ is the most significant level and corresponds to the first rank. The same rule can be applied to convert the *LoA* values from levels to their corresponding ratings for the Channel Security and the Intrusion Response attributes.

### 3. *RLoA Derivation in Elevating Scenarios*

Given that a requester has *n* contextual attributes, ($A_1, A_2, ..., A_n$), and all the attributes are in an *elevating* relationship, and assume that each of the attributes has a confidence value associated to it, ($LoA_{A_1}, LoA_{A_2}, ..., LoA_{A_n}$), then, under the assumption that $LoA_{A_i} > 0$, where $i \in \{1, n\}$, the overall confidence value, *RLoA*, can be calculated (using probability theory) as [27]:

$$RLoA = 1-(1-LoA_{A_1})(1-LoA_{A_2})....(1-LoA_{A_n}) \qquad (4)$$

where $LoA_{A_i}$ is a real value in the interval [0, 1], 1 denoting the highest confidence and 0 the lowest. An advantage of this equation is that an attribute with a higher assurance value would have a higher impact on *RLoA*, and an attribute with a lower assurance value would have a lower impact on the overall assurance value. Applying equation (4) to attributes, *eToken* and *ALoc*, we can calculate $LoA_{authN}$ That is:

$$LoA_{authN} = 1-(1-LoA_{eToken})(1-LoA_{ALoc}) \qquad (5)$$

where the values of $LoA_{eToken}$ and $LoA_{ALoc}$ are given in Tables VI and VII, respectively.

Further applying equations (4) to all the attributes concerned in this section (i.e. *authN, CS and IR)* we obtain an *RLoA* value as:

$$RLoA = 1-(1-LoA_{authN})(1-LoA_{IR})(1-LoA_{CS}) \qquad (6)$$

TABLE V.
OFF-LINE RANK-TO-RATING CONVERSION

| Cardinality | $W_1$ | $W_2$ | $W_3$ | $W_4$ | $W_5$ | ... $W_n$ |
|---|---|---|---|---|---|---|
| 2 | 0.7500 | 0.2500 | | | | |
| 3 | 0.6111 | 0.2778 | 0.1111 | | | |
| 4 | 0.5208 | 0.2708 | 0.1458 | 0.0625 | | |
| 5 | 0.4567 | 0.2567 | 0.1567 | 0.0900 | 0.04 | |
| ... n | | | | | | |

TABLE VI .
ETOKEN RATINGS, *LoA* $_{ETOKEN}$

| Cardinality | Level$_4$ | Level$_3$ | Level$_2$ | Level$_1$ |
|---|---|---|---|---|
| 4 | 0.5208 | 0.2708 | 0.1458 | 0.0625 |

TABLE VII.
ACCESS LOCATION RATINGS, $LoA_{ALoc}$

| Cardinality | Level$_4$ | Level$_3$ | Level$_2$ | Level$_1$ | Level$_0$ |
|---|---|---|---|---|---|
| 5 | 0.4567 | 0.2567 | 0.1567 | 0.0900 | 0.0400 |

It is worth noting that with the *elevating* method, every attribute component *LoA* contributes towards the overall *RLoA*. As a result, the overall *RLoA* will be greater than the maximum *LoA* value afforded by any one of the contextual attributes involved. In some application scenarios or under a certain system setup, Equation (4) may not always be applicable to the attributes of *CS* and *IR*, i.e. equation (6) may not always be true. In such cases, the *weakest-link* method may be more appropriate.

### 4. *RLoA Derivation in Weakest-Link Scenarios*

When the composite influence of multiple contextual attributes follows the *weakest-link* principle, *RLoA* should then be calculated using the *minimum* function. That is, *RLoA* can be calculated using the following formula:

$$RLoA = min(LoA_{authN}, LoA_{IR}, LoA_{CS}) \qquad (7)$$

where *min* is the minimum function that returns the minimum value of those enclosed in the brackets. Note that the calculation of $LoA_{authN}$ remains the same, as the two attributes, *eToken* and *ALoc*, follow the *elevating* relationship due to its two-factor authentication nature.

## IV. CASE STUDY

In this section, the *CRAAC* model is applied to a real-life context-aware authorisation scenario. The scenario describes a Smart Hospital (SH), where Drs. Alice and Bob work. They both have the same organisational role, and use their wireless devices to access the SH restricted services from anywhere (within the hospital). In detail, Alice uses a PDA while Bob uses a wireless laptop. Assuming the patients' data are divided into four categories, denoted by Types$_{1-4}$, and their respective *OLoA* values are given in Table VIII. Users of these services have subscribed to four contextual attributes, namely, *eToken*, *ALoc*, *CS*, and *IR*. Approximately at the same time, Alice and Bob are seeking access to a *Type$_4$* service. Alice is in *Zone$_4$* (i.e. with $LoA_{ALoc}$ of level$_4$) and uses a user-name/password pair as her authentication token. Bob is in *Zone$_1$* (i.e. with $LoA_{ALoc}$ of level$_1$) and has got a PKI smart card. The installed IDS is of a '*High confidence*' type. As both access requests are made at approximately the same time, the $LoA_{IR}$ value associated to both requests corresponds to level$_3$ (i.e. $LoA_{IR}$ level is 3). However, $LoA_{CS}$ varies from Alice to Pop where Alice uses a channel with $LoA_{CS}$=Level$_1$, Bob is on a Level$_3$ channel protection. Table IX summarises the *LoA* values of all these contextual attributes in both levels and ratings (converted using *ROCs*) for both Alice and Bob.

Now, let us compute the *RLoA* for Alice under the assumption the SH is running a strict security policy, i.e. using the *weakest-link* principle. Applying equations (5) and (7), we have $LoA_{authN}$ = 1-(1- 0.1458) (1- 0.4567) = 0.5359 where *RLoA* = *min* (0.5359, 0.2567, 0.0900) = 0.0900. The authorisation decision engine compares Alice's *RLoA,* just computed, to the service Type$_4$ *OLoA* (0.1458). Obviously, as *OLoA*(Type$_4$) > *RLoA*(Alice), which means the assurance level required by the resource *Type$_4$* is higher than what Alice could achieve via her current context information, and therefore Alice is denied access to Type$_4$ services.

However, if the service provider uses an *elevating* security policy, $LoA_{authN}$ remains the same but the overall *RLoA* would be (using equation (6)): $RLoA = 1 - (1-0.5359)(1-0.2567)(1-0.09)=0.686$. In this case, Alice will be granted access to Type$_4$ services as now $OLoA(Type_4) < RLoA$(Alice).

For Bob's access request, when the *weakest-link* security policy is used, *RLoA* is calculated as 0.2567 that is sufficient to grant Bob the access, as $OLoA(Type_4) < RLoA$(Bob). When the *elevating* policy is used, however, Bob's *RLoA* is 0.7591, which will also enable him to access Type$_4$ services.

As shown, *CRAAC* provides the flexibility to allow a service provider to adapt its AC decisions based on the requester's run-time contextual attribute values and the chosen AC policy model (e.g. the *elevated* or the *weakest-link* policies). For instance, if Alice upgrades her channel access software to use a stronger encryption algorithm and crypto key, she would be able to obtain the access permission even if the SH is running the *weakest-link* AC policy. This is because the associated $LoA_{CS}$ value will be increased to 0.4567 (as a result of the channel security upgrade), and *RLoA* using *min* function would produce 0.2567 which is greater than the *OLoA* required by Type$_4$ services.

TABLE VIII.
OLOA REQUIREMENTS

| Service | OLoA | | Description |
|---|---|---|---|
| | Level | Value | |
| **Type$_1$** | 4 | 0.5208 | Patients' DNA data |
| **Type$_2$** | 1 | 0.0625 | Anonymous patient data |
| **Type$_3$** | 3 | 0.2708 | Patients' profiles |
| **Type$_4$** | 2 | 0.1458 | Statistical results for a group of patients |

TABLE IX.
CASE STUDY ATTRIBUTES LOA VALUES

| Context Attribute | Alice's LoA | | Pop's LoA | |
|---|---|---|---|---|
| | Level | Value | Level | Value |
| **eToken** | 2 | 0.1458 | 4 | 0.5208 |
| **Access Location** | 4 | 0.4567 | 1 | 0.0900 |
| **Channel Security** | 1 | 0.0900 | 3 | 0.2567 |
| **Intrusion Response** | 3 | 0.2567 | 3 | 0.2567 |

## V. CONCLUSION AND FUTURE WORK

In this paper we have introduced a new AC model, *CRAAC*, for achieving fine-grained AC in UbiComp environments. Risk assessment and level of assurance play a key role in the *CRAAC* model. Resources are classified into different groups based upon their sensitivity levels and the potential impacts of unauthorised access. Each group is assigned a distinctive *OLoA* denoting the minimum required authorisation level of assurance for that resource group. Upon receiving an object access request, the requester's run-time contextual information is assessed, and an *RLoA* is derived based upon these contextual information. The access request is granted iif $RLoA \geq OLoA$.

*CRAAC* has a number of major advantages over the existing AC approaches. Rather than directly using context information as additional AC constraints, *CRAAC* uses an abstract parameter, the authorisation *LoA*, to decouple the context management from the AC functional module, thus achieving context-aware AC without loosing generality, extensibility and flexibility. Through identifying and grouping users' contextual attributes in relation to authorisation *LoA*, and quantifying and aggregating the contextual information of these attributes into assurance levels, *CRAAC* achieves context-based *LoA* linked AC that allows different contextual information to be captured, and different *LoA* algorithms to be used without affecting the AC module. In other words, through the use of *LoA*, we can have a generic approach to context-aware AC, which can easily be applied to different application domains.

Future work includes designing the architectural components of the *CRAAC* model, and prototyping and evaluating the model to investigate its efficiency and efficacy.

## REFERENCES

[1] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, J. Reitsma. "Context sensitive access control", in *Proc. 10th ACM Symposium on Access Control Models and Technologies (SACMAT '05),* New York, 2005, pp. 111-119.

[2] A. Dey, "Understanding and Using Context", *Personal Ubiquitous Computing, vol. 5(1),* Springer-Verlag, 2001, pp. 4-7,London.

[3] US Office of Management & Budge, "Memorandum M-04-04: E-Authentication Guidance for Federal Agencies" , December, 2003

[4] W. E. Burr, D. F. Dodson, W. T. Polk, "Electronic authentication guideline", *NIST special publication 800-63 version 1.0.2,* April 2006.

[5] R. Sandhu, P. Samarati, "Access control: principles and practice", *IEEE Communications Magazine, vol. 32(9),* 1994, pp. 40-48.

[6] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, "Role-based access control models", *IEEE Computer, vol. 29(2),* February 1996, pp. 38-47.

[7] S-. Chou, "An RBAC-based access control model for object-oriented systems offering dynamic aspect features", *IEICE - Trans. Inf. Syst., vol. 88(9),* Oxford University Press, 2005, pp. 2143-2147.

[8] S-. Park, Y-. Han, T-. Chung, "Context-role based access control for context-aware application". *High Performance Computing and Communications, vol. 4208,* September 2006, Springer Berlin/Heidelberg, pp. 572-580.

[9] M. J. Moyer, M. Ahamad, "Generalized role-based access control",in *Proc. 21st International Conference on Distributed Computing Systems (ICDCS '01),* Washington DC., April 2001, IEEE Computer Society, pp. 391-398.

[10] M. J. Covington, P. Fogla, Z. Zhan, M. Ahamad, "A context-aware security architecture for emerging applications", in *Proc. 18th Annual Computer Security Applications Conference (ACSAC '02),* Washington DC., 2002, pp. 249, IEEE Computer Society.

[11] E. Bertino, P. A. Bonatti, E. Ferrari, "TRBAC: a temporal role-based access control model", *ACM Trans. Inf. Syst. Secur., vol. 4(3),* New York, ACM Press, 2001, pp. 191-233.

[12] S-. Chae, W. Kim, D-. Kim, "Role-based access control model for ubiquitous computing environment", *Information Security Applications, vol. 3786,* February 2006, Springer Berlin / Heidelberg, pp. 354-363.

[13] J. Joshi, E. Bertino, A. Ghafoor, "Hybrid role hierarchy for generalized temporal role based access control model", in *Proc. 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment*

*(COMPSAC '02),* Washington DC., IEEE Computer Society 2002, pp. 951-956.

[14] F. Hansen, V. Oleshchu, "SRBAC: a spatial role-based access-control model for mobile systems", in *Proc. 7th Nordic Workshop on Secure IT Systems (NORDSEC'03).* Gj'vik, Norway 2003, pp. 129-141.

[15] H. Zhang, Y. He, Z. Shi, "Spatial context in role-based access control", *Information Security and Cryptology – ICISC 2006, vol. 4296, November 2006,* Springer Berlin/Heidelberg Lecture Notes in Computer Science 2006, pp. 166-178.

[16] Z. Guangsen, P. Manish, "Context-aware dynamic access control for pervasive applications", in *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, January 2004,* pp. 219-225.

[17] Y-. Kim, C-. Mon, D. Jeong, J-. Lee, C-. Song, D-. Baik, "Context-aware access control mechanism for ubiquitous applications", *Advances in Web Intelligence,* Springer Berlin/Heidelberg, May 2005, vol. 3528, pp. 236-242.

[18] G. H. M. B. Motta, S. S. Furuie, "A contextual role-based access control authorization model for electronic patient record", *IEEE Transactions on Information Technology in Biomedicine, vol. 7(3),* pp. 202-207, 2003.

[19] N. N. Diep, L. X. Hung, Y. Zhung, S. Lee, Y-. Lee, H. Lee, "Enforcing access control using risk assessment", in *Proc. 4th European Conference on Universal Multiservice Networks ( ECUMN '07),* Washington DC., IEEE Computer Society, 2007, pp. 419-424.

[20] K. K. Konrad, T. Konrad, D. David, S. Howard, D. Trevor, "Activity zones for context-aware computing", *UbiComp 2003: Ubiquitous Computing,* Springer Berlin/Heidelberg Lecture Notes in Computer Science, vol. 2864, October 2006, 2003, pp. 90-106.

[21] F. Meneses, A. Moreira, "A flexible location-context representation", in *Proc. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC 2004),* vol. 2, September 2004, pp. 1065-1069

[22] A. Sundaram, "An introduction to intrusion detection", *ACM Crossroads,* vol. 2(4), New York 1996, pp. 3-7.

[23] S. Giles, D. Bersinic, MCSA Windows server 2003 all-in-one exam guide (exams 70-270,70-290,70-291), McGraw-Hill Osborne Media, 2003, pp. 614.

[24] H. Barron, B. Barrett, "Decision quality using ranked attribute weights", *Management Science, vol. 42(11),* November 1996, pp. 1515-1523.

[25] H. Barron, "Selecting a best multiattribute alternative with partial information about attribute weights", *Acta Psychologica, vol. 80,* 1992, pp. 91-103

[26] B. S .Ahn, K. S. Park, "Comparing methods for multiattribute decision making with ordinal weights", *Computers & Operations Research,* Part Special Issue: Algorithms and Computational Methods in Feasibility and Infeasibility, vol. 35(5), May 2008, pp. 1660-1670.

[27] A. Ranganathan, J. Al-Muhtadi, R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments", *IEEE Pervasive Computing, vol. 3(2),* Los Alamitos, IEEE Computer Society 2004 , pp. 62-70.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# User Behaviour Based Phishing Websites Detection

Xun Dong
Department of Computer Science
University of York
York, United Kingdom
Email: xundong@cs.york.ac.uk

John A. Clark
Department of Computer Science
University of York
York, United Kingdom
Email: jac@cs.york.ac.uk

Jeremy L. Jacob
Department of Computer Science
University of York
York, United Kingdom
Email: jeremy@cs.york.ac.uk

***Abstract*—Phishing detection systems are principally based on the analysis of data moving from phishers to victims. In this paper we describe a novel approach to detect phishing websites based on analysis of users' online behaviours – i.e., the websites users have visited, and the data users have submitted to those websites. Such user behaviours can not be manipulated freely by attackers; detection based on those data can not only achieve high accuracy, but also is fundamentally resilient against changing deception methods.**

***Index Terms*—Phishing Attacks, Phishing Websites Detection, Identity Theft, User Protection**

## I. Introduction

PHISHING attacks are well-organised and financially motivated crimes which steal users' confidential information and authentication credentials. They not only cause significant financial damage to both individuals and companies/financial organisations, but also damage users' confidence in e–commerce as a whole. According to Gartner analysts, financial losses stemming from phishing attacks have risen to more than 3.2 billion USD with 3.6 million victims in 2007 in US [23], and consumer anxiety about Internet security resulted in a two billion USD loss in e–commerce and banking transactions in 2006 [22].

The scale and sophistication of phishing attacks have been increasing steadily despite numerous countermeasure efforts. The number of reported phishing web sites increased five–fold from 10047 to 55643 in the 10 month period between June 2006 and April 2007 [3]. The real figure may be much higher because many sophisticated phishing attacks (such as context aware phishing attacks, malware based phishing attacks, and real-time man-in-the-middle phishing attacks against one-time passwords [34]) may not all have been captured and reported.

In this paper we present our design, implementation and evaluation of the *user-behaviour* based phishing website detection system (UBPD). UBPD does not aim to replace existing anti-phishing solutions, rather it complements them. It alerts users when they are about to submit credential information to phishing websites(i.e. when other existing countermeasures fail), and protects users as the last line of defence. Its detection algorithm is independent from how phishing attacks are implemented, and it can easily detect sophisticated phishing websites that other techniques find hard to deal with. In contrast to existing detection techniques based only on the incoming data, this technique is much simpler, needs to deal with much less low level technical detail, and is more difficult to bypass by varying spoofing techniques and deception methods.

Note: In the rest of the paper we use 'interact', 'interaction' and 'user-webpage interaction' to refer to the user supplying data to a webpage.

## II. Related Studies

Researchers have studied the characteristics of phishing attacks [29], [30], [38] and have provided models of phishing attacks [4], [10], [18]. Our understanding of human factors in phishing has been improved by works such as[9], [17], [19], [20], [33]. Novel security interfaces, and automated detection systems have also been invented to protect users.

Anti-phishing email filters can fight phishing at the email level, as it is the primary channel for phishers to reach victims. SpamAssassin [2], PILFER [11], and Spamato [5] are typical examples of those systems. They analyse the incoming emails by applying predefined rules and characteristics often found in phishing emails. PHONEY is a phishing email detection system that tries to detect phishing emails by mimicking user responses and providing fake information to suspicious web sites that request critical information. The web sites' responses are forwarded to the decision engine for further analysis [7].

Web Wallet [40] creates a unified interface for authentication. Once it detects a login form, it asks the user to explicitly indicate the intended site to login. If the intention matches the current site, it automatically fills webpage input fields. Otherwise a warning will be presented to the user.

Phishing website filters in Internet Explorer [24], safe browsing in Firefox 2/3 [27], and Netcraft toolbar [28] are all blacklist anti-phishing websites detection systems. They check whether the URL of the current web page matches any identified phishing web sites before rendering the webpage to users.

SpoofGurad[8] is a signature-and-rule based detection system. It analyses the host name, URL, and the images used in the current webpage to detect the phishing websites. CANTINA [41] uses the TF-IDF information retrieval algorithm to retrieve the key words of the current webpage, and uses Google search results to decide whether the current website is phishy.

Ying Pan et al. have invented a phishing website detection system which examines the anomalies in web pages, in particular, the discrepancy between a web site's identity and its structural features and HTTP transactions [31].

## III. Design

### A. Phishing Nature and Detection Philosophy

Phishing websites work by impersonating legitimate websites, and they have a very short life time. On average a phishing website lasts 62 hours, and users rarely visit the phishing website prior to the attack [26].

Secondly, phishing attacks always generate mismatches between a user's perception and the truth. In successful web based phishing attacks, victims have believed they are interacting with websites which belong to legitimate and reputable organisations or individuals. Thus the crucial mismatch that phishers create is one of real *versus* apparent identity.

Phishing attacks can be detected if we can detect such a mismatch. One approach is to predict a user's perception and then compare it with the actual fact understood by the system. CANTINA is an example of this approach [41]. The main issue with this approach is that the data the system relies on is under the control of attackers, and there are so many techniques that attackers can apply to manipulate the data to easily evade the detection. For CANTINA, attackers could use images instead of text in the body of the webpage, they could use iframes to hide a large amount of content from the users while computer programs can still see it; they could use Javascript to change the content of the page after the detection has been done.

We decide to use another, more reliable, approach. The authentication credentials, which phishers try to elicit, ought to be shared only between users and legitimate organisations. Such (authentication credential, legitimate website) pairs are viewed as the user's *binding relationships*. In legitimate web authentication interactions, the authentication credentials are sent to the website they have been bound to. In a phishing attack the mismatches cause the user to unintentionally break binding relationships by sending credentials to a phishing website. No matter what spoofing techniques or deception methods used, nor how phishing webpages are implemented, the mismatch and violation of the binding relationships always exists. So one can discover the mismatch by detecting violation of users' binding relationships.

Hence phishing websites can be detected when both of the following two conditions are met: 1) the current website has rarely or never been visited before by the user; 2) the data, which the user is about to sumbit, is bound to website other than the current one.

### B. System Design

*1) Overview of the detection work flow:* UBPD has three components:

- **The user profile** contains data to describe the users' binding relationships and the users' personal whitelist. The profile must be constructed before the system can detect phishing websites.

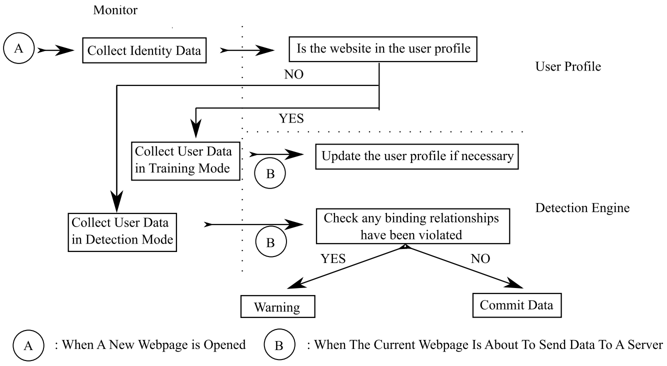

Monitor
A
Collect Identity Data
Is the website in the user profile
User Profile
NO
YES
Collect User Data in Training Mode
B
Update the user profile if necessary
Detection Engine
Collect User Data in Detection Mode
B
Check any binding relationships have been violated
YES
NO
Warning
Commit Data
A : When A New Webpage is Opened
B : When The Current Webpage Is About To Send Data To A Server

Fig. 1. Detection Process Work Flow

- **The monitor** collects the data the user intend to submit and the identity of the destination websites, and activates the detection engine.
- **The detection engine** uses the data provided by the monitor to detect phishing websites and update the user profile if necessary.

UBPD has two working modes: training mode and detection mode. In training mode, UBPD runs quietly in the background, and focuses on learning newly created binding relationships or updating the existing binding relationships. Only in detection mode, UBPD checks whether any of the user's binding relationships would be violated if the user submitted data is sent to the current website. The mode in which UBPD runs is decided by checking whether the webpage belongs to a website

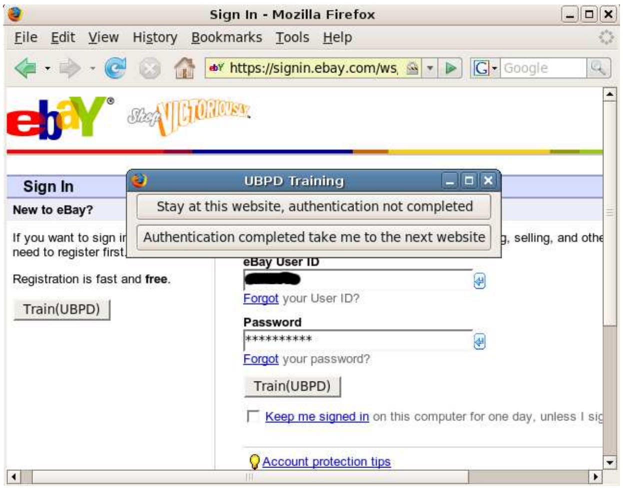
1) whose top level domain[1] is in the user's personal whitelist, or
2) with which the user has shared authentication credentials.

If either is true the system will operate in training mode, otherwise, in detection mode. Phishing webpages will always cause UBPD to run in the detection mode, since they satisfy neither condition.

The detection work flow is shown in Figure 1. Once a user opens a new webpage, the monitor decides in which mode UBPD should be running. Then, according to the working mode the monitor chooses appropriate method to collect the data the user submitted to the current webpage, and sends it to the detection engine once the user initiates data submission. The details of the data collection methods are discussed in section IV. When running in detection mode if the binding relationships are found to be violated, the data the user submitted will not be sent and a warning dialogue will be presented. For the remaining cases, UBPD will allow the data submission.

*2) Creation of the User Profile:* The user profile contains the user's binding relationships and personal whitelist. The binding relationships are represented as a collection of paired records, i.e., $\langle aTopLevelDomain, aSecretDataItem \rangle$. The personal whitelist is a list of top level domains of websites.

[1] Suppose the URL of a webpage is "domain2.domain1.com/files/page1.htm", the top level domain is "domain1.com"


Sign In - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://signin.ebay.com/ws
Google
ebay
Shop VICTORIOUSLY.
UBPD Training
Stay at this website, authentication not completed
Authentication completed take me to the next website
Train(UBPD)
Train(UBPD)

Sign In

New to eBay?

If you want to sign in, need to register first.

Registration is fast and free.

eBay User ID

[redacted]

Forgot your User ID?

Password

**********

Forgot your password?

□ Keep me signed in on this computer for one day, unless I sig

Account protection tips

Fig. 2. User Profile Creation

To make sure the binding relationship records are created, UBPD makes it as a mandatory step of UBPD's installation procedure. Having installed the UBPD, when the browser is restarted for the first time, the user is presented with a dialogue, which asks for the websites with which the user has accounts. Then UBPD automatically takes users to those websites, and asks users to fill in the authentication forms on those websites one by one. In addition, UBPD also dynamically modifies each authentication webpages, so that all the buttons on the webpage will be replaced with the 'Train(UBPD)' button. Once a user clicks on it, UBPD creates new binding relationship records in the user profile and then asks whether the user wants to carry on the authentication process (in case it is a multiple step authentication) or go to the next website. A screen shot is shown in Figure 2.

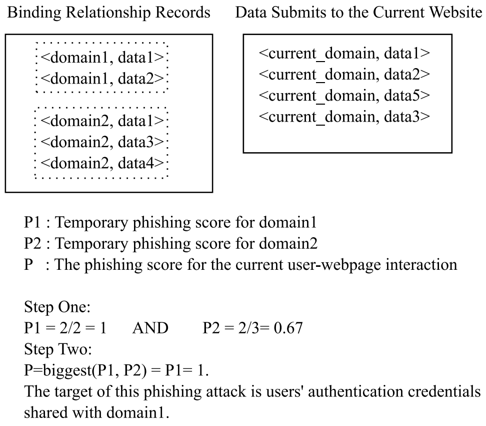
On average users have over 20 web accounts [12]. We do not expect them to train UBPD with all their binding relationships, however, we do expect them to train UBPD with their most valuable binding relationships (such as their online accounts with finanical organisations). Our preliminary evaluations have confirmed that our assumption is valid.

There are two types of binding relationships that UBPD is unaware of: the ones that already exist but users have not trained UBPD with, and the ones that users will create in the future. It is possible that the authentication credentials in these unknown binding relationships have also been used in the binding relationships that UBPD is aware of (as the result of authentication credentials reuse). When this is the case, with the binding relationships records alone UBPD interprets the legitimate use of the unknown binding relationships as a violation of binding relationship, and wrongly raise a phishing attack warning. To avoid such false warnings, the whitelist is automatically created to include the websites that users have already got accounts with but did not make UBPD aware of and the websites they may have account with in future. As stated in section III-B1 if the website with which the user is currently interacting is found to be in the whitelist UBPD will only run in training mode. In the training mode, such reuse of the authentication credentials can only cause UBPD to either create new binding relationships record or update existing ones.

The personal whitelist is constructed by combining a default whitelist with the websites the user has visited more than three times (configurable) according to the user's browsing history. The default whitelist is constructed by identifying the 1000 websites that are most visited in the user's country. This information is obtained from Alexa, a company specialising in providing internet traffic ranking information. The default whitelist could be viewed as a reflection of mass users' online browsing behaviours. These most popular websites are not phishing websites, and users are likely to have binding relationships with them in the future if they do not have already. In this way, websites that users currently have accounts with or are going to have account with are very likely to be included in the whitelist. By default, this whitelist is automatically updated weekly. It is worthy noting that, the whitelist in UBPD is mainly used for reducing false warnings, it is not used as conventional whitelist to decide whether the current website is legitimate or not.

*3) Update of the User Profile:* Besides the manual update by users, when running in the training mode UBPD has an automatic method to update the user profile with the other unknown binding relationships. It detects whether the user is using a web authentication form. This is achieved by analysing the HTML source code, such as the annotation, label, use of certain tags (such as ⟨form⟩) and type of the HTML elements. If the user is using a web authentication form, and the user profile contains no binding relationships with the current website, then UBPD prompts a window to ask the user to update the user profile. If there is an existing binding relationship for the current website, then UBPD will replace the authentication credentials in the binding relationships with the lastest values the user submits. If users have entered the authentication credentials wrongly, those credentials will still be stored, but those wrong values will be corrected, when users relog in with the correct authentication credentials. In future the detection of web authentication page usage can be much simpler and more accurate once web authentication interfaces [15], [35], [37] are standardised. Current authentication form detection is still accurate because it deals with legitimate websites (automated update is carried out only in training mode), and those legitimate websites use standard web forms and will not deliberately disrupt detection.

*4) Phishing Score Calculation:* In detection mode, UBPD decides whether the current webpage is a phishing webpage by calculating phishing scores. The calculation is a two step process. In the first step, for each legitimate website, with which the user has shared authentication credentials, a temporary phishing score is calculated. Each temporary phishing score is the fraction of the authentication credentials associated with a legitimate website that also appear in the data to be submitted to the current webpage. Its value ranges from $0.0$ to $1.0$.



P1 : Temporary phishing score for domain1
P2 : Temporary phishing score for domain2
P : The phishing score for the current user-webpage interaction

Step One:
P1 = 2/2 = 1 AND P2 = 2/3= 0.67
Step Two:
P=biggest(P1, P2) = P1= 1.
The target of this phishing attack is users' authentication credentials shared with domain1.

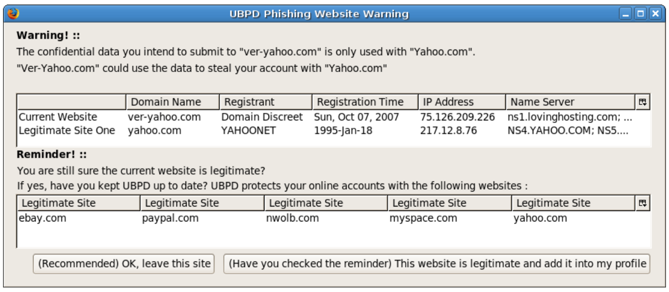
Fig. 3. An Example of How Phishing Score Is Calculated

In the second step, those temporary phishing scores are sorted into descending order. The current webpage's phishing score is the highest score calculated. The legitimate website with the highest temporary phishing score is considered to be the impersonated target of the phishing website. If more than one legitimate website has yielded the highest phishing score (due to the reuse of authentication credentials), they will all be considered as targets. Although it may not be the attacker's intent, the data they get if an attack succeeds can certainly be used to compromise the user's accounts with those legitimate websites. Figure 3 illustrates how a phishing score is calculated in UBPD.

Given our phishing score calculation method, clever attackers may ask the user victims to submit their credential information through a series of webpages, with each phishing webpage asking only for a small part of data stored in the user profile. To handle this fragmentation attack UBPD has a threshold value and cache mechanism. Once the phishing score is above the threshold the current webpage will be considered as a phishing webpage. The system's default threshold is $0.6$. Why we choose $0.6$ is discussed in section V. If the current phishing score is not zero, UBPD also remembers which data has been sent to the current website and will consider it in the next interaction if there is one. Accordingly many fragmentation attacks can be detected.

*5) Reuse:* It is very common that a user shares the same authentication credentials (user names, passwords, etc) with more than one website. The two running modes and the user's personal whitelist are designed to prevent false warnings caused by reuse without compromising detection accuracy.

UBPD detects the violation of binding relationships only when the user is interacting with websites that are neither in the user's whitelists nor which the user has account with. So as long as legitimate websites for which users have used the same authentication credentials are all contained in the user profile, there will be no false phishing warning generated due to the reuse. The method that UBPD uses to create the user profile ensures such legitimate websites are most likely to be included, as those websites are either within the user's browsing history or are popular websites in the user's region. Our preliminary false positive evaluation also confirmed this.



**Warning! ::**
The confidential data you intend to submit to "ver-yahoo.com" is only used with "Yahoo.com".
"Ver-Yahoo.com" could use the data to steal your account with "Yahoo.com"

| | Domain Name | Registrant | Registration Time | IP Address | Name Server |
|---|---|---|---|---|---|
| Current Website | ver-yahoo.com | Domain Discreet | Sun, Oct 07, 2007 | 75.126.209.226 | ns1.lovinghosting.com; ... |
| Legitimate Site One | yahoo.com | YAHOONET | 1995-Jan-18 | 217.12.8.76 | NS4.YAHOO.COM; NS5.... |

**Reminder! ::**
You are still sure the current website is legitimate?
If yes, have you kept UBPD up to date? UBPD protects your online accounts with the following websites :

| Legitimate Site | Legitimate Site | Legitimate Site | Legitimate Site | Legitimate Site |
|---|---|---|---|---|
| ebay.com | paypal.com | nwolb.com | myspace.com | yahoo.com |

(Recommended) OK, leave this site
(Have you checked the reminder) This website is legitimate and add it into my profile

Fig. 4. Phishing Warning Dialogue

*6) Warning Dialogue:* The content of the warning dialog must be suitable for users with very limited knowledge, otherwise, as previous user study [39] found, users may ignore the warning or may not behave as suggested. Figure 4 is a warning dialog example. To make the information easy to understand, the dialogue tells users that the current website, to which they are submitting credentials, is not one of the legitimate websites associated with those authentication credentials. To help users understand the detection result and make a correct decision, UBPD also provides information regarding the differences between the legitimate website and the possible phishing website in five aspects: the domain name, the domain registrant, the domain registration time, name servers, and IP addresses. Users don't need to understand those terms. They only need be able to recognise the difference between values of the five attributes of the legitimate website and the phishing website.

*7) Website Equivalence:* To discover whether the user is about to submit the authentication credentials to entities, with which they have not been bound, UBPD needs to be able to accurately decide whether a given website is equivalent to the website recorded in user's binding relationships. It is much more complicated than just literally comparing two URLs or IP addresses of two websites, because: 1) big organisations often have web sites under different domain names, and users can access their account from any of these domains; 2) the IP address of a website can be different each time if dynamic IP addressing is used; 3) it is hard to avoid 'pharming' attacks, in which the phishing site's URL is identical to legitimate one.

Our system first compares two websites' domain names and IP addresses. When the two domain names and two IP addresses are equal the web sites are assumed to be identical. Otherwise, the system interrogates the WHOIS[2] database and uses the information returned to determine equivalence. When analysing two different IP addresses our system compares the netnames, name servers, and the countries where each IP address is registered. If they are all identical then the two websites are deemed to be identical. This method can

[2] WHOIS is a TCP-based query/response protocol which is widely used for querying a database in order to determine the owner of a domain name, an IP address. RFC 3912 describes the protocol in detail.

also detect pharming attacks, in which both fraudulent and legitimate websites have the same domain name but are hosted on different IP addresses.

This method is not perfect. A more elegant and complete solution would be a mechanism where servers provide security relevant metadata (including the outsourcing information) to the web browser via a standard protocol as suggested by Behera and Agarwal [6]. However, unless it becomes a standard we have to rely on WHOIS. The extended validation certificate [13] can provide information to decide whether two websites belong to the same party, but the problem is the high cost, high entry level and complexity of obtaining such certificates. Many small and medium businesses will not be able to own them; as a result they cannot be used to decide website equivalence in general.

*8) User's Privacy:* Since a user profile contains confidential user information, it is important that it does not add new security risks. We use a one-way secure hash function to hash the confidential data before it is stored. When the system needs to determine the equivalence between the data, the system just needs to compare the hash values.

However, because the domain name of the website is not hashed, if the profile is obtained by attackers they would be able to find out with which websites users have accounts and where users have reused their authentication credentials. This information is helpful for attackers; for example, it can be used to launch context aware phishing attacks against the users. To prevent this, when the system is installed it will randomly generate a secret key, and this key will be used to encrypt and decrypt the domain names.

### C. Implementation

UBPD is implemented as a Firefox add-ons. Most of the detection is implemented using Javascript. The hash function we use is SHA-1[21] and the encryption method we use is Twofish[1]. The hash function and encryption methods can be changed, we choose to use them mainly because there are open source implementations available. The user interface of the system is implemented by using XUL, a technology developed by Mozilla [14]. Although only implemented on Firefox, the detection system can be easily developed to work on other browsers, such as Internet Explorer and Opera.

## IV. Evasion and Countermeasures

### A. Disruption of Data Collection

To accurately detect phishing UBPD needs to be able to work with the exact data the user is submitting to the current webpage. This data can be easily retrieved by accessing the DOM interface, however, the client script language can also manipulate the data before the monitor retrieves it.

In training mode, websites that users are visiting are legitimate, the monitor uses the DOM interface to retrieve the data, as in those cases client scripts manipulation is not an issue. However, in detection mode, when the user is interacting with websites that may be phishing websites, client scripts manipulation is a real threat. The monitor takes hence a different approach. The monitor performs like a keylogger, it collects user data by listening to a user's keystrokes for each element in the webpage. Such key logging is very efficient to run and since UBPD runs in detection mode only occasionally, there is little performance overhead.

### B. Activation of the detection engine & denial of service attack

The detection engine must be activated before the data the user entered has been sent to a remote server. Normally webpages forward the user submitted data using built-in functions, such as the one provided by the standard web form (Almost every legitimate website uses this function to submit the data). So the detection engine is triggered and data submission is suspended when the monitor discovers the use of such functions. This is achieved by listening for 'DOMActivate' [16] events. These events are fired by the browser when such built-in functions have been used. This is the mechanism the monitor uses to activate the detection engine when the system is running in training mode.

However, phishers can use client side scripts to implement these built in functions. For example, by using Javascript they can access the DOM interface to retrieve the data users entered and use AJAX (Asynchronous Javascript And XML) techniques to send the data to the server before the user clicks the submit button. To prevent this evasion, UBPD could monitor the client script function calls when it is running under the detection mode. The function calls should be monitored are 'open()' and 'send()' from the 'xmlhttprequest' API [36]. These are the only two functions that client scripts can use to send data to the server. Once the monitor discovers such function calls, the function is suspended and the detection engine is triggered. Regardless of the data the client scripts would like to send, the detection engine always works on all the data the user has entered to the current webpage. The function call is only resumed if the detection engine thinks the current website is legitimate.

Since the detection engine would be triggered once the 'xmlhttprequest' API has been called, the attackers can issue this function call continously to freeze the browser. To prevent this, the monitor can keep a record of whether there is user input since the last time the detection engined was activated for the same webpage. If there is then the detection engine will be activated, otherwise, not.

We plan to implement such client side scripts function call detection by using the techniques in Venkman (a Firefox JavaScript Debugger Addon) in the next version of UBPD. Nevertheless, if attacked by this evasion technique, at least UBPD would still be able to inform users what they have done, and preserve the evidence for possible analysis. One of the major problem for users is they do not know when they have been attacked and how. It gives users a good chance to reduce the damage caused by the phishing attacks.

## V. Evaluation

We have carried out two experiments to evaluate the effectiveness of UBPD in terms of the two following rates:

TABLE I
CHARACTERISTICS OF USER PROFILE

| | Alice | Bob | Carol | Dave |
|---|---|---|---|---|
| Reuse | No | No | Yes | Yes |
| Uniqueness | Strong | Weak | Strong | Weak |

- **False negative**: The system fails to recognise a phishing attack.
- **False positive**: The system recognises a legitimate website as a phishing website.

In addition we also search for a useful default threshold value (mentioned in section III-B4). For both experiments UBPD was modified to not present the warning dialogue, instead it records the phishing score results as well as the URLs for later analysis.

### A. *False Negative Rate*

From PhishTank[32] and millersmiles[25] we randomly collected 463 recently reported phishing webpages, which target Ebay, Paypal, and Natwest bank. We created four user profiles, which describe four artifical users' binding relationships with the three targeted websites. The four user profiles have different characteristics as shown in Table I. 'Reuse' indicates maximum possible reuse of authentication credentials. In this case the user would have same user name and password for Ebay and Paypal. 'Uniqueness' indicates whether the user would use the exact data they shared with a legitimate website at other places. For example if Bob chooses his email address as password then the uniqueness is weak, because Bob is very likely to tell other websites his email address. If Bob uses some random string as his password, then the uniqueness is strong, because this random string is unlikely to be used with any other websites.

We entered the artifical authentication credentials to each phishing webpages. Regardless of the characteristics of the user profile, the detection result is the same for all four users: 459 pages had a phishing score of 1, and 4 had a phishing score of 0. Thus only four pages evaded detection – a 99.14 percent detection rate. Compared to other existing phishing website detection systems, UBPD's detection rate may not be significantly better. **Its biggest advantage** is that its detection method detects essential characteristics of a phishing attack, namely that phishing web pages request authentication credentials. The details of how users may be manipulated may change with future phishing attacks, but the requesting of such details remains constant. Other detection systems based on the analysis of incoming data will need to adapt and be redesigned for future phishing attacks; UBPD will not.

Detailed analysis confirms that the detection result is determined mainly by the information requested by the phishing webpage. Table II shows the classification of the phishing webpages based on the type of information they requested. 92% of the collected phishing webpages asked only for authentication credentials and 7.14% of the collected phishing webpages asked both for personal and authentication credentials.

TABLE II
STATISTICS OF PHISHING WEBSITES BASED ON THE TYPE OF INFORMATION REQUESTED. AC: AUTHENTICATION CREDENTIALS; PI: PERSONAL INFORMATION

| | Ebay | Paypal | Natwest |
|---|---|---|---|
| AC | 211 | 176 | 39 |
| AC+PI | 22 | 5 | 6 |
| PI | 4 | 0 | 0 |
| Total | 237 | 181 | 45 |

The four phishing web pages that UBPD failed to detect asked only for personal information such as full name, address, telephone number and mother's maiden name. In fact, they can not be detected by UBPD no matter what the threshold value is. However, it is impractical for phishing attacks to ask for personal information without requesting authentication credentials first, because those phishing webpages are not presented to users when a user victim first arrives at the phishing website. Those phishing websites would normally first present the user with a login webpage before directing the user to the webpage that asking for the personal information (none of the four phishing webpages were the landing page of the phishing attacks). Otherwise such practice would be seemed abnormal, and make potential victims very suspicious. As a result, UBPD can detect the phishing attacks and stop users from even reaching the phishing webpages that ask for personal information.

The sample size in this experiment is large and we might have some expectation that this would be reasonably indicative of success rate when deployed 'in the wild'.

### B. *False Positive Rate*

Five volunteers were provided with the information needed to install UBPD on their machine. We did not explictly ask them to train UBPD with all their binding relationships, because we wanted to see how users would train UBPD and what the false positives would be in reality if the user has not properly trained UBPD. At the end of one week, we collected the result log from their machines.

The volunteers were three male and two female science students. They all used Firefox as their main web browser. They were all regular Internet users (in average over three hours per day). As a result the UBPD was activated a large number of times and the interactions that occurred during the experiments covered a wide range of types of interaction. Another reason we chose those volunteers is because they are the most unlikely user group to fall victims to phishing attacks [17] and so we can safely assume they have all interacted with legitimate websites. In total the volunteers interacted with 76 distinct websites, sumbitted data to those websites 2107 times, and UBPD ran in detection mode only 81 times. In fact all the websites volunteers visited were legitimate. On 59 occasions the phishing score was $0$, on five interactions gave a score of $0.25$, on 13 occasions the score was $0.5$, and the score was $1$ on three occasions.

The phishing score was 1 when users interacted with three legitimate websites (the registration webpages of videojug.com and surveys.com, and the authentication webpage of a web forum). We asked the volunteers what data they supplied to those webpages. It seems that the reuse of authentication credentials on creating new accounts is the reason. In this experiment, the warning dialog is not presented, as we did not aim to test usability. But if it does, then the user must make decision to train UBPD to remember these new binding relationships. To avoid the user's confusion about what is the right choice when the warning dialog is presented, the dialog always reminds the user of the legitimate websites UBPD is aware of, and tells the user that if the user is sure the current website is legitimate, and the website is not remembered by UBPD, then they need to update their binding relationships (see the figure 4 in section III-B6). This requires no technical knowledge and should be quite easy to understand. There are only two choices provided by the dialog: to update the profile and submit the data; or donot send the user submitted data and close the phishing webpage. There is no third choice provided by the dialog, in this way we force the user to make the security decision and they can not just ignore the warnings given by the system.

Many websites force users to supply an email address as the user name. As a result, the user's email address is kept in the user profile as part of user's authentication credentials. This email address almost inevitably will be given out to other websites, which are not contained in the user profile, for various reasons such as contact method, activate the newly registered account, etc. Thus even when the user does not intend to give out their credentials, the email address nevertheless is shared and our system simply confirmed that by calculating the phishing score of $0.5$ (which means half of the data the user has shared with a legitimate website was given away to a website that was not in user's profile) on 13 occasions.

For one volunteer five interactions gave a phishing score of $0.25$. The user had an account at a major bank, the authentication credentials for which compromised four data items. One of these was the family name. For other sites not included in the user's profile asking for this information caused our system to identify the sharing of the data.

Based on the figures from both experiments we decided to set the default threshold value to $0.6$. First, it can successfully detect phishing webpages asking for more than half of the credentials the user has shared with a legitimate website (99.14% of the phishing websites in Experiment One can be detected). It also generated few false positives. The false positive rate of the system is $0.0014$ (obtained by dividing the number of false positives generated with the total number of times the UBPD was activated).

Of course this is only a preliminary false positive evaluation, but it still shows the system has a small false positive rate, and it also shows that the reuse of the authentication credentials and partial training are the main cause of the false positives.

## VI. Discussion

### A. Why UBPD is useful

Besides its high detection accuracy, UBPD is useful also because it complements existing detection systems. First UBPD detects phishing websites based on users' behaviours, not the incoming data that attackers can manipulate freely. Violation of the binding relationships cannot be changed no matter what techniques phishers choose to use. As the evaluation proves, UBPD is able to consistently detect phishing webpages regardless of how they are implemented as long as they ask for authentication credentials. In contrast detection systems based on incoming data may find it difficult to deal with novel and sophisticated spoofing techniques. UBPD analyses the identity of websites using both IP addresses and domain names, it can detect pharming attacks, which are undetectable by many existing systems. Being independent of the incoming data means low cost in maintenance, the system does not need updating when attackers vary their techniques, and so we have far fewer evasion techniques to deal with.

Some systems that try to stop phishing attacks from reaching the users (phishing site take down, botnet take down, phishing email filter, etc.), some have tried to detect phishing webpages as soon as users arrive at a new web page (Phishing URL blacklist, netcraft toolbar, spoofguard, CARTINA, etc), and some have tried to provide useful information to help users to detect phishing websites. However, there are no other systems that work at the stage when phishing webpages have somehow penetrated through and users have started to give out information to them. Thus our system plugs a major remaining gap, and it can be easily combined with other phishing detection systems to provide multi-stage protection.

### B. Performance

The two working modes of UBPD reduce the computing complexity. Computing in detection mode is more expensive, but it runs only occasionally (in our second experiment UBPD only in detection mode 81 out of 2107 times). Computing in the detection mode is still light weight for the computing power of an average personal computer (none of the volunteers noticed any delay). As a result, UBPD is efficient to run and adds few delay to the existing user-webpage interaction experience.

### C. Limitations and Future Work

UBPD should be viewed as an initial but promising effort towards detecting phishing by analysing user behaviour. Despite its detection effectiveness, there are some limitations within the implementation and design.

Currently UBPD cannot handle all types of authentication credentials. It can handle static type authentication credentials such as user name, password, security questions, etc, but dynamic authentication credentials shared between users and the legitimate websites cannot be handled by the current implementation (e.g. one-time passwords). In future we will investigate how to handle such credentials.

There is also another method for a phisher to evade detection. Attackers might compromise a website within a user's personal whitelist and host the phishing website under the website's domain (perhaps for several hours to a couple of days). This method is difficult to carry out, and it is unlikely to happen if there are still easier ways to launch phishing attacks.

Detection itself requires little computing time, in fact, retrieving the data from WHOIS database is the most time consuming part. It depends on the type of network the user is using and the workload on the WHOIS database. In future we could add some cache functionality to reduce the number of queries for each detection. Currently the system produces a slight delay because of the WHOIS database query.

Although we have paid attention to the usability design of the system, we have not systematically evaluated it. Further user evaluation will inform development.

## VII. Acknowledgement

I wish to thank all the people helped us, and especially the Engineering and Physical Sciences Research Council(EPSRC) of United Kingdom, without their sponsorship of the project "Defending the weakest link: Intrusion via Social Engineering" the detection method would not be invented.

## VIII. Conclusion

We have used a novel paradigm—analysis of the users' behaviours—to detect phishing websites. We have shown that it is an accurate method, discussed how it has been designed and implemented to be hard to circumvent, and have discussed its unique strength in protecting users from phishing threats. UBPD is not designed to replace existing techniques. Rather it should be used to complement other techniques, to provide better overall protection. We believe our approach fills a significant gap in current anti-phishing technology capability.

## References

[1] http://home.versatel.nl/MAvanEverdingen/Code/.

[2] Spamassassin's official website. http://spamassassin.apache.org/.

[3] Anti-phishing work group home page, 2007. http://www.antiphishing.org/.

[4] C. Abad. The economy of phishing: A survey of the operations of the phishing market. *First Monday*, 10(9), 2005.

[5] K. Albrecht, N. Burri, and R. Wattenhofer. Spamato—An Extendable Spam Filter System. In *2nd Conference on Email and Anti-Spam (CEAS), Stanford University, Palo Alto, California, USA*, July 2005.

[6] P. Behera and N. Agarwal. A confidence model for web browsing. In *Toward a More Secure Web—W3C Workshop on Transparency and Usability of Web Authentication*, 2006.

[7] M. Chandrasekaran, R. Chinchain, and S. Upadhyaya. Mimicking user response to prevent phishing attacks. In *IEEE International Symposium on a World of Wireless, Mobile, and Multimedia networks*, 2006.

[8] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell. Client-side defense against web-based identity theft. In *NDSS '04: Proceedings of the 11th Annual Network and Distributed System Security Symposium*, February 2004.

[9] R. Dhamija, D. Tygar, and M. Hearst. Why phishing works. *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems,*, ACM Special Interest Group on Computer-Human Interaction:581–590, 2006.

[10] X. Dong, J. A. Clark, and J. Jacob. A user-phishing interaction model. In *Conference on Human System Interaction*, 2008.

[11] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 649–656, New York, NY, USA, 2007. ACM Press.

[12] D. Florencio and C. Herley. A large-scale study of web password habits. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 657–666, New York, NY, USA, 2007. ACM Press.

[13] R. Franco. Better website identification and extended validation certificates in ie7 and other browsers. IEBlog, November 2005.

[14] B. Goodger, I. Hickson, D. Hyatt, and C. Waterson. Xml user interface language (xul) 1.0. Technical report, Mozilla Org., 2001.

[15] S. Hartman. Ietf-draft: Requirements for web authentication resistant to phishing. Technical report, MIT, 2007.

[16] B. Hhrmann, P. L. Hgaret, and T. Pixley. Document object model events. Technical report, W3C, 2007.

[17] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *ACM Communication*, October 2007.

[18] M. Jakobsson. Modeling and preventing phishing attacks. In *Phishing Panel in Financial Cryptography '05*, 2005.

[19] M. Jakobsson. Human factors in phishing. *Privacy & Security of Consumer Information '07*, 2007.

[20] M. Jakobsson, A. Tsow, A. Shah, E. Blevis, and Y.-K. Lim. What instills trust? a qualitative study of phishing. In *Extended abstract, USEC '07*, 2007.

[21] P. A. Johnston. http://pajhome.org.uk/crypt/index.html.

[22] A. Litan. Toolkit: E-commerce loses big because of security concerns. Technical report, Garnter Research, 2006.

[23] T. McCall. Gartner survey shows phishing attacks escalated in 2007; more than $3 billion lost to these attacks. Technical report, Gartner Research, 2007.

[24] Microsoft. Anti-phishing white paper. Technical report, Microsoft, 2005.

[25] MillerSmiles. Official website. http://www.millersmiles.co.uk.

[26] T. Moore and R. Clayton. Examining the impact of website take-down on phishing. In *eCrime '07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 1–13, New York, NY, USA, 2007. ACM.

[27] Mozilla. Phishing protection, 2007. http://www.mozilla.com/en-US/firefox/phishing-protection/.

[28] Netcraft, 2007. http://toolbar.netcraft.com/.

[29] G. Ollmann. The phishing guide. Technical report, NGSS.

[30] G. Ollmann. The pharming guide. Technical report, Next Generation Security Software Ltd., 2005.

[31] Y. Pan and X. Ding. Anomaly based web phishing page detection. *acsac*, 0:381–392, 2006.

[32] Phishtank, 2007. http://www.phishtank.com/.

[33] S. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies. In *2007 IEEE Symposium on Security and Privacy*, 2007.

[34] R. Security. Enhancing one-time passwords for protection against real-time phishing attacks. Technical report, RSA, 2007.

[35] G. Staikos. Web browser developers work together on security. Web, November 2005.

[36] A. van Kesteren. The xmlhttprequest objec. Technical report, W3C, 2006.

[37] W3C. Web security context—working group charter. Web, 2006.

[38] D. Watson, T. Holz, and S. Mueller. Know your enemy: Phishing. Technical report, The Honeynet Project & Research Alliance, 2005.

[39] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610, New York, NY, USA, 2006. ACM Press.

[40] M. Wu, R. C. Miller, and G. Little. Web wallet: preventing phishing attacks by revealing user intentions. pages 102–113, 2006.

[41] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 639–648, New York, NY, USA, 2007. ACM Press.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 791 – 798

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A New Worm Propagation Threat in BitTorrent: Modeling and Analysis

Sinan Hatahet
Heudiasyc UMR 6599
Université de Technologie de Compiègne
BP 20529
60205 Compiègne cedex
France
Email sinan.hatahet@utc.fr

Abdelmadjid Bouabdallah
Heudiasyc UMR 6599
Université de Technologie de Compiègne
BP 20529
60205 Compiègne cedex
France
Email boubdallah@ut c .fr

Yacine Challal
Heudiasyc UMR 6599
Université de Technologie de Compiègne
BP 20529
60205 Compiègne cedex
France
Email yacine.challal@utc.fr

*Abstract*—**Peer-to-peer (p2p) networking technology has gained popularity as an efficient mechanism for users to obtain free services without the need for centralized servers. Protecting these networks from intruders and attackers is a real challenge. One of the constant threats on P2P networks is the propagation of active worms. Recent events show that active worms can spread automatically and flood the Internet in a very short period of time. Therefore, P2P systems can be a potential vehicle for active worms to achieve fast worm propagation in the Internet. Nowadays, BitTorrent is becoming more and more popular, mainly due its fair load distribution mechanism. Unfortunately, BitTorrent is particularly vulnerable to topology aware active worms. In this paper we analyze the impact of a new worm propagation threat on BitTorrent . We identify the BitTorrent vulnerabilities it exploits, the characteristics that accelerate and decelerate its propagation, and develop a mathematical model of their propagation. We also provide numerical analysis results. This will help the design of efficient detection and containment systems.**

## I. Introduction

PEER-TO-PEER systems, like eMule, BitTorrent, Skype, and several other similar systems, have became immensely popular since the past few years, primarily because they offered a way for people to get a free service. According to Androutsellis et al. [1] “Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance, without requiring the intermediation or support of a global centralized server or authority”. Under the hood, these systems represent a paradigm shift from the usual web of client and servers, to a network where every system acts as an equal peer. Moreover, due to the huge number of peers, objects can be widely replicated, therefore increasing the availability of the provided services, despite the lack of centralized infrastructure. This leads to the proliferation of a variety of applications, examples include multicast systems, anonymous communications systems, and web caches. P2P systems consume up to 70 % of the Internet overall traffic[4].

The ease of use, provided services and finally the low price; all contribute in the increasing number of P2P users. However this fact also inspired attackers to attack P2P networks. Making these systems "secure" is a significant challenge. Indeed, a malicious node might give erroneous responses to a request, both at the application level (returning false data to a query, perhaps in an attempt to censor the data) or at the network level (returning false routes, perhaps in an attempt to partition the network). BitTorrent, on the other hand, supplies means of checking the integrity of the data received upon its reception, thus protecting the users from file poisoning and similar attacks. However, BitTorrent fails to prevent attacks that are designed to induce an impact on the network’s infrastructure. These attacks may cause damages of great consequences on the quality of service and the reliability of ISPs. Such attacks are similar to distributed denial of service (DDoS) attacks.

There are several attacks that can be conducted against BitTorrent. Such attacks are like S ybil attacks [6], Eclipse attacks [7], DDoS attacks, or even active worms attacks. Active worms are programs that self-propagate across the Internet by exploiting security flaws in widely-used services [8].

In this paper we analyze the impact of a new worm propagation model on BitTorrent. BitTorrent is particularly vulnerable to topology aware active worms. Topology aware worms use the topologic information found on their victims to find new victims. Such worms are capable of quickly flooding the Internet while escaping current deployed intrusion detection systems. Moreover, in order to boost its initial propagation the worm uses a trackers’ hitlist consisting of the most crowded swarms (*i.e.* groups of BitTorrent users interested in the same content). This mechanism allows the worm to find newer victims even faster than traditional scanning worms. This combination of both scanning strategies (*i.e. the strategy a worm uses to discover new machines to infect*) is fatal, because it provides the worm with certainty, discretion and speed. This combination allows the worm to only attack existing targets, thus saving scanning time and more importantly escaping the current implemented detection systems, since such systems are normally installed on non-attributed addresses.

The possible damages that such worm can causes are huge, our analysis of its propagation shows that it can achieve a 300% increase in its propagation speed in comparison with traditional scanning worms. The purpose of our research is to further investigate this new worm, identify the characteristics that accelerate and decelerate its propagation in BitTorrent, and to develop a mathematical model of their propagation. Such model would be used to compare the worm behavior in different scenarios and thus, better identify its weaknesses and strengths. We believe that our work can provide important guidelines for P2P system design and control to address the concerns of active worms and to develop efficient containment and intrusion detection systems. The rest of the paper is organized as follows. In section 2, we first discuss how the BitTorrent works in practice and then discuss "Tit-for-Tat" algorithm in general. In section 3, we present related and previous research done on P2P worms. In section 4, we explicate the strategy of our developed topology aware worm (the BitTorrent worm). In section 5, we present our model, and provide numerical analysis results. We end up this paper with conclusion and future work in section 6.

## II. BitTorrent

BitTorrent is a P2P protocol for content distribution and replication designed to quickly, efficiently and fairly replicate data [3]. Recent reports have indicated that near 75 % of all the current P2P Internet traffic is due to BitTorrent (see figure 1) [4]. In contrast to other P2P protocols, the BitTorrent protocol does not provide any resource query or lookup functionality, but rather focuses in fair and effective replication and distribution of data. BitTorrent works by groups of users, called *swarms*, with the interest of downloading a single specific file, coordinating and cooperating to speed-up the process.

A *swarm* can be partitioned into two network entities: a *tracker*, and peers [8]:

1. A *tracker* is a centralized software which keeps track of all peers interested in a specific file. Each *swarm* is managed by a *tracker* .
2. The second entity is the set of active peers, which can be further divided into *seeds* and *leeches* . A *seed* is defined as a peer that has already retrieved the entire shared file. Where a *leech* is a downloading peer.

A server, usually a web server is also important for the smooth conduct of BitT*orren*t. The *purpo*se *of th*is server is to provide a *torrent* file for interested clients. The *torrent* file i*s a f*ile that contains the necessary information for the clients to prepare the download and join the *swarms*. The main information in this file is a set of (SHA-1, [9]) hash values, which allows the user to verify the integrity of the received file content. The file stores the address of the *tracker* as well. In this paper, we give enough relevant details to allow us to facilitate the description of the attack.

In figure 1, we illustrate how a client downloads a file from a BitTorrent *swarm*. *Leeches* are represented in a red color, while *seeds* are represented in green. The *tracker* is installed on a machine which is located in a *swarm* represented by a cloud. Let's imagine a scenario, where a client shows an interest in downloading a certain file. The client first searches for the desired file by consulting a known website (see figure 1 step 1). The client would then downloads a *torrent* file which its metadata matches the desired file (see figure 1 step 2). Next, the client will read the content of the torrent file, and get the tracker address (see figure 1 step 3). Once, the client obtains the tracker address, he gets connected to it, announces its will to download the shared file and asks the tracker about other peers (see figure 1 step 4). When asked for peers, a tracker will return a random list of other peers currently in the swarm. As the number of peers in a single swarm may become very large for popular files, the size of the returned list is usually bound; a maximum of 50 peers is typical (see figure 1 step 5) [8]. Once a client has obtained a list of other peers, it will contact them to try to fetch the data it is looking for.

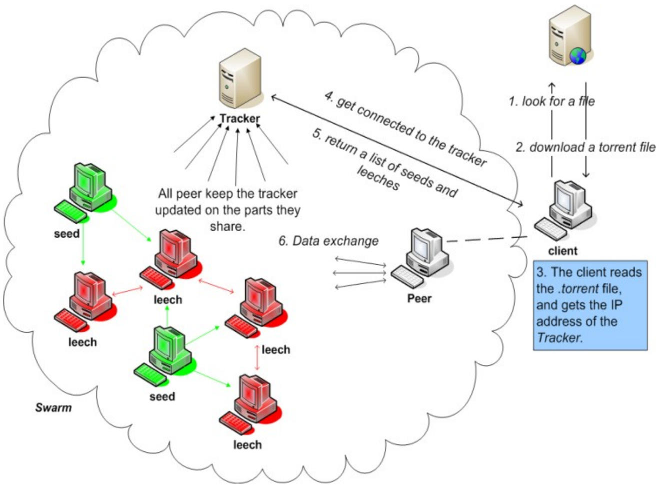

1. look for a file
Tracker
4. get connected to the tracker
5. return a list of seeds and leeches
2. download a torrent file
All peer keep the tracker updated on the parts they share.
seed
6. Data exchange
client
3. The client reads the .torrent file, and gets the IP address of the Tracker.
leech
Peer
leech
leech
seed
Swarm
leech

Figure 1: How BitTorrent work

The bandwidth being a limited resource, a single client cannot serve every peer interested in pieces it holds at the same time. The maximum number of peers served concurrently (i.e. the number of available slots) is configurable by the user and depends on the available bandwidth. All other peers connected to a client (whether they are interested or not) which are not being served are said to be choked . In consequence, each client implements an algorithm to choose which peers to choke and un-choke among those connected to him over time. The strategy proposed by BitTorrent is named "tit-for-tat", meaning that a client will preferably cooperate with the peers cooperating with him. Practically, this means that each client measures how fast it can download from each peer and, in turn, will serve those from whom it has the better download rates. When a client has finished downloading a file, it no longer has to download from other Peers but it can still share (upload) pieces of the file. In this case the choking algorithm is applied by considering upload rate instead. Peers are selected based on how fast they can receive the upload. This spreads the file faster. Such "seeder" peers that store the whole file are very important to the functioning of a swarm. If a swarm contains no seeders it may lead to a situation in which pieces of the file are missing from the swarm as a whole. In this sense, the system requires some level of altruistic behavior from "seeders". This behavior is encouraged by the matra often repeated on BitTorrent websites: leave your download running for a little while after you have got the entire file [8].

## III. P2P Worms

A computer worm is a program that propagates itself over a network, reproducing itself as it goes [10]. Due to its recursive nature, the spread rate of a worm is very huge and poses a big threat on the Internet infrastructure as a whole. The purpose of a worm is to achieve a high infection rate within the targeted hosts (*i.e.* infects the largest number possible of vulnerable machines). Modern worms may control a substantial portion of the Internet within few minutes. No human mediated response is possible to stop an attack that is so fast. The possible devastating effects on the Internet operation are hard to underestimate. It was reported in the FBI/CSI survey, that in 2007 52% of the detected network attacks were viruses' attacks (worms/spyware). Moreover, they caused damages worth the amount of 8,391,800 USD in the United States alone [11]. Besides the traffic generated by the worm propagation is so huge that it can be considered as a DDoS attack on the whole Internet and could be used to bring down the Internet infrastructure of whole countries. Therefore a huge number of researches were carried out in order to conceive proper detection and containment systems. However, there is a new trend of worms that is emerging and which have a huge destruction potential, such worms are called Peer-to-Peer worms. A P2P worm is a worm that exploits the vulnerabilities of a P2P network in order to propagate itself over the network and accelerate its propagation throughout the Internet. P2P worms could be much faster than the old-fashion worms. Furthermore they are expected to be one of the best facilitators of Internet worm propagation due the following reasons: [8] [6] [12] [13] [14]

*i)* P2P systems have a large number of registered active hosts which easily accelerate Internet worm propag ation, as hosts in P2P systems are real and active;

*ii)* some hosts in P2P systems may have vulnerable network and system environments, e.g., home networks;

*iii)* Hosts in P2P systems maintain a certain number of neighbors for routing purposes. Thus, infected hosts in the P2P system can easily propagate the worm to their neighbors, which continue the worm propagation to other hosts and so on.

*iv)* they are often used to transfer large files,

*v)* the programs often execute on user's desktops rather than servers, and hence are more likely to have access to sensitive files such as passwords, credit card numbers, address books…etc

*vi)* The use of the P2P network often entails the transfer of "grey" content (e.g., pornography, pirated music and videos), arguably making the P2P users less inclined to draw attention to any unusual behavior of the system that they perceive.

In order to identify the characteristics of worms, we need to understand how it propagates itself over a network. A typical worm works as follows: it first scans the Internet to find potential victims (*i.e.* information collection). Once it locates a machine the worm tries to probe it by exploiting a common vulnerability, if successful it transfers a copy of its malicious code to the new victim and so on. The key of a successful worm is its propagation speed rather that the vulnerability it exploits. Since current deployed detection and containment systems are capable of blocking the spread of relatively slow worms, a worm should propagate quickly, regardless the vulnerability it is exploiting, in order to achieve a high infection rate. Choosing an efficient scanning strategy enables the worm to reach a large population in a record time.

Based on the scanning strategies of P2P worms, they could be classified into two broad categories: passive worms and active worms (see figure 5). Passive worms are identical to viruses in the sense that they do not search for new victims, they however await them. On the other hand, active worms search for vulnerable targets. Indeed, active worms are more dangerous and propagate faster than passive worms.

### *A Passive worms:*

A passive worm does not spread in an automated fashion. However, it stays dormant on infected machines, waiting for other vulnerable machines to reach it. Once a connection is established between a vulnerable machine and the infected one, the worm duplicates itself on the other end and infects it. This kind of worms can be developed to exploit the vulnerabilities of any Internet application.

### *B. Active worms:*

Active worms propagate by infecting computer systems and by using infected computers to spread the worms in an automated fashion [12]. We can classify P2P active worms into two categories: *Hitlist worms* which attack a network using a pre-constructed list of potential vulnerable machines; and the *topologic worms* which attack a network based on the topologic information found on their victims.

## IV. The Bittorrent Worm

In this section we will explain a novel propagation strategy of a BitTorrent worm and compare it with previous work.

### *A. Background:*

In a previous study [6], Yu et al. presented a propagation scheme for the Topologic P2P worm. In this strategy, after joining the P2P system at the system's initial time, the infected host immediately initiates an attack against its P2P neighbors with its full attack capacity. If extra attack capacity is available, the infected hosts would randomly attack the Internet. This propagation strategy is unfortunately not realistic, since the worm attacks only its neighbors at the initial instant of its infection, and does not seek future neighbors that will connect to it later. Hence, the worm achieves a low infection rate within the P2P system's peers. Furthermore the authors avoided creating cooperation between infected hosts for simplicity. Therefore, victims could be attacked by different infected hosts, and at multiple times during the attack.

### *B Overview:*

The main idea of our novel propagation model is to increase continuously the number of overlay neighbors of infected peers in order to speed up the topologic worm propagation. Our model uses the concept of *"honey pot"*, where an infected peer advertises itself as a seed in a **popular** swarm. Hence, infected hosts will rapidly attract new victims in the popular swarms. A hit-list of popular swarms is shared among infected peers to increase the number of *"honey pots"*, and thus accelerate the propagation.

### *C. The BitTorrent Worm:*

The BitTorrent Worm (BTW) is a Topology aware worm. Accordingly, like the topologic worms, as soon as a new host is infected by the BTW, it starts attacking its overlay neighbors. If extra attack capacity is available, BTW does not just sit around and waits for new peers to fall in its trap, but goes as far as advertising himself in order to attract new peers. BTW is capable of doing so, by joining new crowded swarms and announcing itself as a *seed*. This is possible since the Tracker does not check the integrity if new comers. Once it joined the swarms, new leeches will automatically try to connect to it. Furthermore, unlike the Topologic worm, BTW does not ignore attacking peers whom will later connect to it. Hence, BTW tends to reach a larger population inside in the BitTorrent network.

Another major limitation of the Traditional Topologic worm was the lack of cooperation between its instances, BTW overrides this constraint by enforcing two levels of cooperation on the infected hosts. The first one is on the *swarm* level and the other one is on the BitTorrent network level. The cooperation on the *swarm* level is achieved upon the time of infection. Once an infected host succeeds in infecting a new victim, it passes a list of the peers it scanned to the victim, so the newly infected host does not waste its attack capacity in re-attacking them. As for the cooperation on BitTorrent level, it is achieved as follows: the attacker builds a Hitlist of trackers responsible for the most crowded swarms and then provided it to the initial worm instance. The worm-infected hosts will continuously join the swarms on the list and start attacking their members. Upon the time of infection, the infected host will communicate half of its list to its victim and so on. Once the Trackers Hitlist exhausted the infected hosts would randomly attack the Internet. The Hitlist should be sorted with regards to the population of swarms to boost the initial propagation of the worm. The detailed algorithm is as follows:

*Algorithm 2:*

```
1. P = 0 // list of peers to infect
   C : attack capacity
   Cr = C // remaining attack capacity
   Cu = 0 //used attack capacity
   i = 0 // last index retrieved in tracker Hitlist

 N = 0 //list of neighbors
2. While (true)
       N = new neighbors
     if  (not empty(N))
        Cu = min (Cr , capacity to attack(N))
        P = peers to attack (Cu)
        startAttack(P)
        Cr = C – Cu
        if (i < HT.length AND Cr > 0)
              join swarm at HT(i)
              i = i +1
     if (Cr > 0)
        randomAttack(1)

startAttack(N)
   for each n in N
       if (n is vulnerable and non-infected)
           I passes HT/2 to n
           I passes the list of the nodes it scanned to n
           // so n does not waste its C scanning them again
           Cr = Cr +1
   end
end startAttack(N)

randomAttack (hosts)
   j = 0
   while ( j < hosts.length)
       randomly choose n
       if (n is vulnerable and non-infected)
            I passes the list of the nodes it scanned to n
           Cr = Cr +1
         j = j+1
   End while
end randomAttack (hosts)
```

In figure 2, we illustrate how BTW propagates through BitTorrent. *Leeches* are represented in a red color, while *seeds* are represented in green and *infected hosts* in black. For the sake of simplicity we consider that the attack capacity of the worm is 5. The attack starts as follows: once the malicious code is installed on initial infected machine *I*, *I* starts attacking its neighbors {1, 2, 3, 4}. (We assume in this example that *I* infects its neighbors orderly). At the time of infection, *I* would pass half of its tracker Hitlist to its victim 1, in order to share its workload with 1. Furthermore *I* would also pass its list of scanned hosts to 1, in order to coordinate their efforts. *I* would repeat the procedure upon the infection of its neighbors 4 and 3. In the other hand, since 2 is not vulnerable, *I* cannot infect it. However, since *I* has already scanned it, *I* would still pass its address to its victims so they do not scan it once more. Since *I* has 4 neighbors only, it uses the remaining of its attack capacity in exploiting its *Trackers Hitlist* (*i.e.* announces itself as a *seed* to the *tracker* of the *swarm* #2 on its *Hitlist*).

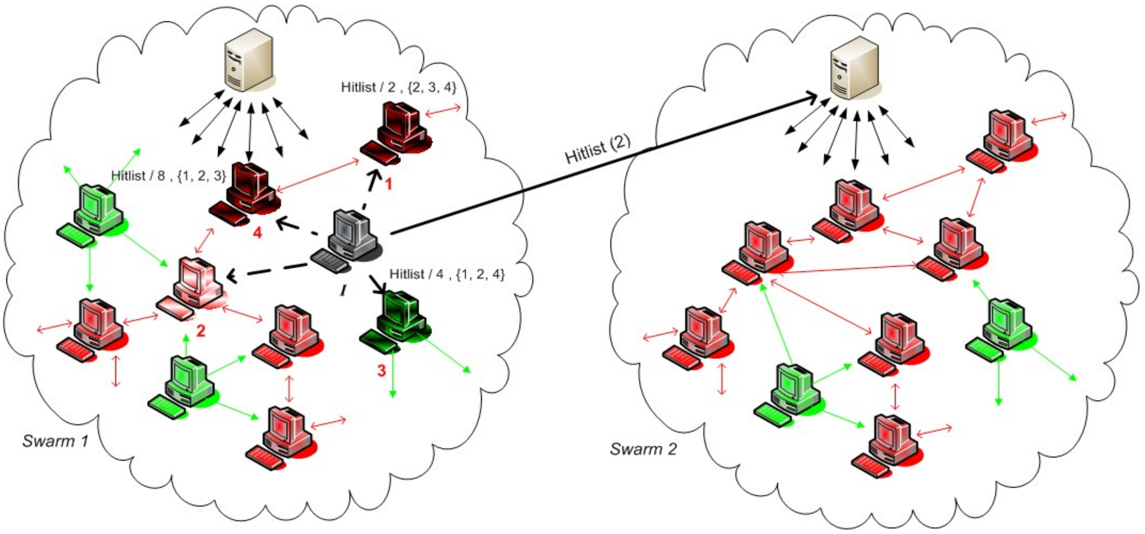


**Figure 2 : BTW attack**

Once *I* advertises itself as a seed in swarm 2, new leeches will automatically try to connect to it. If at instant *t*, a new node connects to *I*, *I* attacks it instantly. Moreover, the previously infected nodes {1, 3, 4} follow the example of *I,* waits for new neighbors, and respectively join swarms from their part of the Hitlist.

## V. Modeling BTW

Modeling worm propagation is very important because it allows us understanding how they evolve; whom they would reach and how long they take to contaminate the network. Moreover, modeling allows us to identify which parameters play a role in their propagation and therefore develop proper and efficient detection and containment mechanisms.

### *A. Parameters*

In order to formally build a model that describes the BTW propagation, we need to identify the different factors play a role in its propagation. After a thorough examination and analysis we identified that the following parameters, which will have an impact on the worm propagation.

*1) Attacker parameters:* The attack capacity of the worm and the system's initial infected worm instances are the most important parameters from the worm attacker perspective. Intuitively, the larger these values are, the faster the propagation is.

*2) P2P system parameters:* For P2P-based systems, the following parameters need to be considered:

*i)* The topology degree of P2P systems: the average nu mber of neighbors connected to each peer.

*ii)* The size of P2P system: defines the number of hosts in a P2P system.

*iii)* The number of peers within a swarm.

*iv)* The vulnerability of P2P systems: measures the vuln erability of P2P hosts. As mentioned before, a host in a P2P system could be used in less protected environments, such as a home environment.

The join and leave rate of BitTorrent peers: defines the number of peers that respectively join and leave BitTo rrent.

*3) The Internet parameters:* these parameters are imposed by the nature of the Internet as well as the presence of detection systems.

The join and leave rate of Internet hosts: defines the number of hosts that respectively join and leave the Internet.

*i)* The average of Internet connection speed.

*ii)* The patch rate, t he rate at which an infected or vulnerable machine becomes invulnerable .

*iii)* The death rate, the rate at which an infection is detected on a machine and eliminated without patching.

### *B. Assumptions*

We assume that the IP system address space is the IP address space of IPv4, thus 2 . In the IPv4 address space, some valid IP addresses are not actively utilized, are non-routable, or are even not applicable to the host (based on previous statistical result [6] , only 24% of available addresses are used by active hosts). We assume that there are two logical systems: a "P2P" system, which represents BitTorrent in the Internet. The other is called "nonP2P" system; it represents the rest of Internet. In both "P2P" system and "nonP2P" system, we assume that a number of hosts are vulnerable. As our analysis considers the average case, we assume that each host in "P2P" or "nonP2P' system has a certain probability to be vulnerable. In this paper, we do not consider the time taken for the infected host to find the vulnerability of victims and assume that the worm infects one victim within one unit time. At the system's initial time, we assume that there are a certain number of infected hosts and infected hosts are already in the "P2P" system. We assume that the join and leave rates are uniform. Furthermore, we assume that the average speed of connection of each peer is 240 kBps [16] , that the size of a BitTorrent packet is 64 kB [15] , and that the number of addresses returned by a tracker upon a request is 50 peers [8].

In table 1, we summarize the different notation used in the description of our model:

TABLE 1:
NOTATIONS IN THIS PAPER

| Parameters | Notations |
|---|---|
| ***T*** | Total IP addresses in the system |
| $S_i$ | Size of “P2P” system at instant *i* |
| $P_v$ | Proportion of vulnerable hosts in the Inte rnet |
| ***C*** | Attack capacity of worm infection host (number of victims being able to be scanned simultaneously) |
| $\lambda_{aP2P}$ | The rate at which a peer joins BitTorrent |
| $\lambda_{lP2P}$ | The rate at which a peer leaves BitTorrent |
| $\lambda_{conn}$ | The number of downloading request r eceived by peer per second. |
| $\lambda_{BTT}$ | The rate at which a peer downloads a Bi tTorrent packet. (i.e. = average connection speed 240 KBps / size of a BitTorrent packet 64KB ) |
| $\lambda_A$ | The rate at which a hosts joins the Internet |
| $\lambda_L$ | The rate at which a hosts leaves the Internet |
| $\lambda_P$ | The rate at which an infected or vulnerable machine becomes invulnerable |
| $\lambda_D$ | The rate at which an infection is detected on a machine and eliminated without patc hing |
| ***V(i,P2P)*** | The number of vulnerable hosts in P2P at the time *i* ( *V* (0, *P2P* ) is the number of vulnerable hosts which can be infected at the system initial time = $S_0$ * $P_v$ ) |
| ***V(i,I)*** | The number of vulnerable hosts in nonP2P at the time *i* ( *V* (0, *I* ) is the number of vulnerable hosts which can be infected at the system initial time = *T* * 0.24 * $P_v$ ) |
| ***I(i,P2P)*** | The number of infected peers in P2P at the time *i* ( *I* (0, *P2P* ) is the number of initial infected hosts in the system.) |
| ***I(i,I)*** | The number of infected peers in nonP2P at the time *i* ( *I* (0, *I* ) is the number of initial infected hosts in the system.) |
| ***I(i,ALL)*** | The total number of infected hosts at the time *i* |
| ***newI(i, P2P)*** | The number of newly infected hosts in P2P added at step *i* ( *newI* (0, *P2P* ) = 0) |
| ***newI(i,I)*** | The number of newly infected hosts in non-P2P added at step *i* ( *newI* (0, *I* ) = 0) |
| ***CacheSize*** | The number of peers a peer can simultaneously upload to. |
| $N_j(i)$ | The number of neighbors , *j* can attack at instant *i* |
| $P_{add}$ | The probability of the address of a peer in BitTorrent is returned by a tracker upon request of resources. |
| $avg_{peers}$ | The average number of peers in a swarm |
| $avg_{leeches}$ | The average number of leeches in a swarm ( $avg_{leeches} = avg_{peers}$ * *0.83 ) [16] |
| ***num*** | The number of peers in a swarm |

*C. Model*

To better understand the characteristics of the BTW spread, we adopt the epidemic dynamic model for disease propagation. In order to make it flexible for analyzing BTW, we use discrete time to conduct recursive analysis and approximate the worm propagation [6] [17]. In what follows, we will calculate the number of infected peers by BTW at instant *i* : *I(i,ALL)*.

***Lemma 1*** **:** The size of BitTorrent evolves as follows:

$$S_{i+1} = S_i * (1 + \lambda_{aP2P} - \lambda_{lP2P}) \tag{1}$$

*Proof* : The size of BitTorrent (1) increments by the number of infected peers which joined BitTorrent at the instant *i* , and decremented by the number of infected peers which left BitTorrent at the instant *i*.

***Lemma 2*** **:** The number of neighbors to scan of each peer in BitTorrent evolves as follows:

```
foreach $I(i+1, P2P)$
    $avg_{leeches} = \frac{V(i,P2P) - I(i,P2P)}{S_i / avg_{peers}}$
    $P_{add} = \frac{50}{avg_{peers}}$
    $\lambda_{conn} = P_{add} * [\lambda_{aP2P} + (avg_{leeches} * \lambda_{BTT})]$
    if ( $C < N_I(i)$ )
        $N_I(i+1) = min(N_I(i) - C + \lambda_{conn}, cache\ size)$
    else
        $N_I(i+1) = min(\lambda_{conn}, cache\ size)$          (2)
```

*Proof* : The number of neighbors $N_I(i+1)$ (2) increases by the number of exchanging request received , and decreases by the number of *peers* the worm scanned at instant *i.* The num ber of neighbors a peer can have in BitTorrent is represented by the size of its uploading\downloading cache (*i.e.* the number of simultaneously maintained connections). The number of downloading request received per second $(\lambda_{conn})$ , is the number of peers which has been redirected by a tracker at instant *i*. In BitTorrent, a *leech* is redirected to another peer upon its request for resources. A *leech* usually asks for resources, when it first joins the swarm, and when it finishes downloading a BitTorrent packet. Hence, $\lambda_{conn}$ is the sum of $\lambda_{aP2P}$ and $avg_{leeches} * \lambda_{BTT}$ multiplied by the probability of being redirected by a tracker upon a request of resources $P_{add}$ .

***Proposition 1*****:** In the ‘P2P’ system, with *V(i,P2P), I(i,P2P),* $S_i$ and $N_I$ ( *i* ) at time *i* , the next tick will have

$$newI(i+1, P2P) = [V(i,P2P) - I(i,P2P)] * \left[1 - \left(1 - \frac{1}{S_i}\right)^{\sum_{j=1}^{I(i,P2P)} \min[N_j(i), C]}\right] \tag{3}$$

*Proof* : The number of newly infected peers in BitTorrent (3): is the number of vulnerable but not infected peers (*i.e.* $[V(i,P2P) - I(i,P2P)]$) by the probability of being scanned by infected peers (*i.e.* $[1 - (1 - 1/T) \wedge \sum_{j=1}^{I(i,P2P)} \min(N_j(i), C)\ ]$ ).

$$I(i+1, P2P) = [I(i,P2P) * (1 - \lambda_{lP2P} - \lambda_P - \lambda_D)] + newI(i+1, P2P) \tag{4}$$

The number of infected peers in BitTorrent (4): is therefore, incremented by the number of newly infected machine in [1], and decremented by the number of patched and detected infected peers, and the number of infected peers which left BitTorrent at the instant *i*.

***Proposition 2*:** In the ‘nonP2P’ system, with *V(i,I), I(i,I)* and $N_I$ ( *i* ) at time *i* , the next tick will have

$$newI(i+1,I) = [V(i,I) - I(i,I)] * \left[1 - \left(1 - \frac{1}{T}\right)^{I(i,ALL)*C - \sum_{j=1}^{I(i,P2P)} \min[N_j(i),C]}\right] \tag{5}$$

*Proof* : The number of newly infected hosts over the Internet and outside BitTorrent (5): is the number of vulnerable but not infected hosts ( *i.e.* $[V(i,I) - I(i,I)]$ ) by the probability of being scanned by infected peers (*i.e.* $[1-(1-1/T)^\wedge I(i,ALL)*C - \sum_{j=1}^{I(i,P2P)} \min(N_j(i),C)]$ ).

$$I(i+1,I) = [I(i,I)*(1-\lambda_L-\lambda_P-\lambda_D)] + newI(i+1,I) \tag{6}$$

The number of infected hosts in the Internet and outside BitTorrent (6): is therefore, incremented by the number of newly infected machine in (5), and decremented by the number of patched and detected infected peers and the number of infected peers which left BitTorrent at the instant *i*.

***Corollary*:** In all the “Internet”, with *I(i,P2P)* and *I(i,I)* at time *i* , the next tick will have

$$I(i+1,ALL) = I(i+1,I) + I(i+1,P2P) \tag{7}$$

The number of infected hosts all over the Internet is (7): the sum of the number of infected peers in BitTorrent (4), and the number of infected hosts outside BitTorrent (6).

## V. Numerical Results

In this section, we evaluate the numerical performance by using models with different parameters for different scenarios. We report the performance results along with observations.

### *A. Simulation Model:*

- *Metrics:* For each of the scenarios, the system attack performance is defined as follows: the time taken *t* (X axis) to infected host number (Y axis). The higher the performance value, the worse is the attack effect.
- *Parameters :* The general system is defined by the tuple: <*A* , *T, C,* $S_0$ , $P_v$ , $\lambda_A$ , $\lambda_L$ , $\lambda_{BTT}$ , $\lambda_P$ , $\lambda_D$ , *I(0,P2P), CacheSize, num* >, representing the system configuration parameters. *A* determines the attack strategy and can be one of < *BTW, Topologic, Random scanning* >. Other parameters are explained in Table 1. As we are only focusing on selected important parameters that are sensitive to BTW, the following parameters are set with constant values ( *T*= $2^{32}$ , *C=6, I(0,P2P) = 5,* $\lambda_{BTT}$ *= 3.75* ) in all our simulations.

### *B. Perfor mance Results:*

In this section, we report the performance results along with observations.

*1) Impact of the attack strategy* : Fig. 3 illu *st* rates the sensitivity of attack performance depend *ing on different attack strateg* ies. The general system is configured as <* , 2, 6 , 1 * *1*, 0.2 , 0.01, 0.01 , 3.75 , 0.00002 , 0.00002 , 5 , 30 , 2000>. We notice that the BTW attack strategy outperforms the tradition *al* Top *ogic at tack str ategy* as well as t *rando* m scanning attack strategy . For example, in the worm fast propagation phase (linear increase – from simulation time 40 to 85), the BTW approach can achieve 300% performance increase over the Topologic attack strategy. The result matches our expectation: achieving a higher infection rate in the P2P system significantly improves the attack performance. From the defense perspective, the BTW attack will be a very challenging issue.

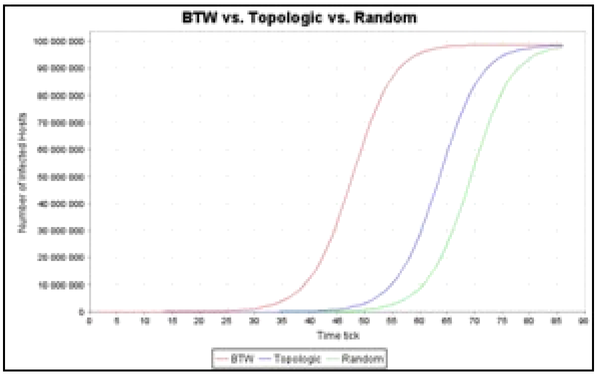


Figure 3. Perormance comparison ofAll Attack St rat egies

*2) The impact of P2P System Size:* Fig. 4 illustrates the sensitivity of BTW performance under different sizes of BitTorrent network. The general system is configured as <BTW, 2, 6, *, 0.2, 0.01, 0.01, 3.75, 0.00002, 0.00002, 5, 30, 2000>. In this figure, the size of BitTorrent varies in $S_0${2* 1, 4* 1, 1* 1}. We notice that increase of BitTorrent network size enhances the attack performance. The result matches previous observations [6] [18]: the larger is the size of the P2P system, the higher is the achieved scan hit probability.

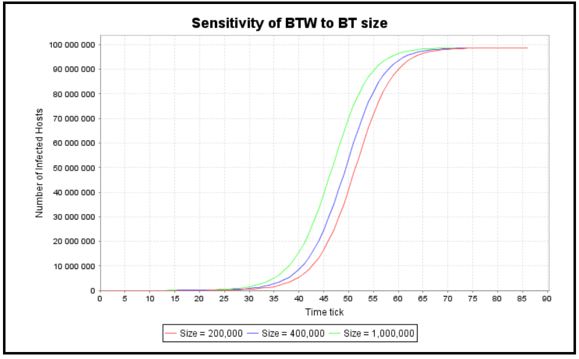


**Figure 4: The Sensitivity of BTW to BitTorrent Size**

*3) The impact of P2P Topology Degree:* Fig. 5 illustrates the sensitivity of BTW performance within the P2P system for different BitTorrent topology degrees. This is represented by the limit of topologic neighbors a peer can have (i.e. cache size).The general system is configured as <BTW, 2, 6, 1* 1, 0.2, 0.01, 0.01, 3.75, 0.00002, 0.00002, 5, *, 2000>. In this figure, the Y axis represents the number of infected peers in BitTorrent. We notice that an increase in topology degree achieves better attack performance. This matches our expectation; a larger topology degree makes more P2P hosts open to BTW and speeds up the worm propagation.

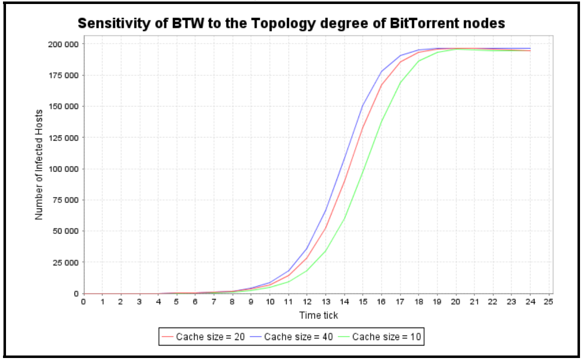


**Figure 5: The Sensitivity of BTW to the Topology degree of BitTorrent nodes**

## VI. CONCLUSION

In this paper we analyzed the impact of a novel worm propagation model on BitTorrent. BitTorrent is particularly vulnerable to topology aware active worms. Topology aware worms use the topologic information hold by their victims to find new victims. Such worms are capable of quickly flooding the Internet while escaping current deployed intrusion detection systems. Moreover, in order to boost its initial propagation the worm uses a trackers' hitlist consisting of the most crowded swarms. This mechanism allows the worm to find newer victims even faster than traditional scanning worms. This combination of both scanning strategies is fatal, because it provides the worm with certainty discretion and speed. Our analysis of this propagation scheme shows that it can achieve a 300% increase in its propagation speed in comparison with traditional scanning worms. W e developed a mathematical model to describe this new propagation strategy, and provided numerical analysis results. We believe that our work provides important guidelines for P2P system design and control that address the concerns of active worms and to develop efficient containment and intrusion detection systems.

## REFERENCES

1. *A Survey of Peer-to-Peer Content Distribution Technologies.* Androutsellis-Theotokis, S. and Spinellis, D. s.l. : ACM Computing Surveys, 2004., 2004.
2. *A Survey of Peer-to-Peer Security Issues.* Wallach, D.S. Tokyo, Japan : Springer, November 8-10, 2002.
3. *Incentives build robustness in BitTorrent.* Cohen, B. May 2003.
4. *P2P survey 2007.* Ipoque.
5. *A measurement study of piece population in BitTorrent.* C, Dale et J, Liu. Washington DC : GlobeCom, November 26–30 2007.
6. W. Yu, C. Boyer, S. Chellappan, D. Xuan. Peer-to-peer system-based active worm attacks: Modeling and analysis. *IEEE International Conference on Communications (ICC).* May 2005.
7. C. Göldi, R. Hiestand. *Scan Detection Based Identification of Worm Infected Hosts.* Zurich : Swiss Federal Institute of Technology, , 18 April 2005. ETHZ.
8. Hales, D and Patarin, S. *How to cheat bittorrent and why nobody does.* s.l. : Department of Computer Science University of Bologna, May 2005. TR UBLCS-2005-12.
9. *Measurement and Analysis of BitTorrent Signaling Traffic.* Erman, David, et al. Oslo : NTS17, 2004.
10. Joukov, N. and Chiueh, T. Internet worms as internet-wide threat. *Experimental Computer Systems Lab, Tech. Rep. TR-143, September.* 2003.
11. Richardson, Robert. *2007 CSI Computer Crime and Security Survey.* s.l. : Computer Security Institute, 2007.
12. Abhishek, Sharma et Vijay, Erramilli. Worms: attacks, defense and models. s.l. : Computer Science Department, University of Southern California.
13. http://www.us-cert.gov/cas/tips/ST04-015.html. *CERT.*
14. Nassima Khiat, Yannick Carlinet, Nazim Agoulmine. The Emerging Threat of Peer-to-Peer Worms. *MonAM 2006 Workshop.* 2006.
15. Bittorrent Protocol Specification v1.0. *Theory.org.* http://wiki.theory.org/BitTorrentSpecification.
16. Pouwelse, J.A., et al. The bittorrent p2p file-sharing system: Measurements and analysis. *International Workshop on Peer-to-Peer Systems (IPTPS).* 2005.
17. Chen, Z. S., Gao, L.X. et Kwiat, K. Modeling the Spread of. *In Proceedings of IEEE INFOCOM, San Francisco.* March 2003.
18. Tao, Li, Zhihong, Guan and Wu, Xianyong. Modeling and analyzing the spread of active worms based on P2P systems. *Computers & Security* . Issue 3, May 2007, Vol. Volume 26, Pages 213-218.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Information System Security Compliance to FISMA Standard: A Quantitative Measure

Elaine Hulitt
U. S. Army Engineer Research and
Development Center, CEERD-ID-S
3909 Halls Ferry Road
Vicksburg, MS 39180-6199, USA
Email:Elaine.Hulitt@usace.army.mil

Rayford B. Vaughn, Jr., Ph.D.
Department of Computer Science and Engineering
Center for Computer Security Research
P.O. Box 9637, Mississippi State University
Mississippi State, MS 39762, USA
Email:Vaughn@cse.msstate.edu

***Abstract*—To ensure that safeguards are implemented to protect against a majority of known threats, industry leaders are requiring information processing systems to comply with security standards. The National Institute of Standards and Technology Federal Information Risk Management Framework (RMF) and the associated suite of guidance documents describe the minimum security requirements (controls) for non-national-security federal information systems mandated by the Federal Information Security Management Act (FISMA), enacted into law on December 17, 2002, as Title III of the E-Government Act of 2002. The subjective compliance assessment approach described in the RMF guidance, though thorough and repeatable, lacks the clarity of a standard quantitative metric to describe for an information system the level of compliance with the FISMA-required standard. Given subjective RMF assessment data, this article suggests the use of Pathfinder networks to generate a quantitative metric suitable to measure, manage, and track the status of information system compliance with FISMA.**

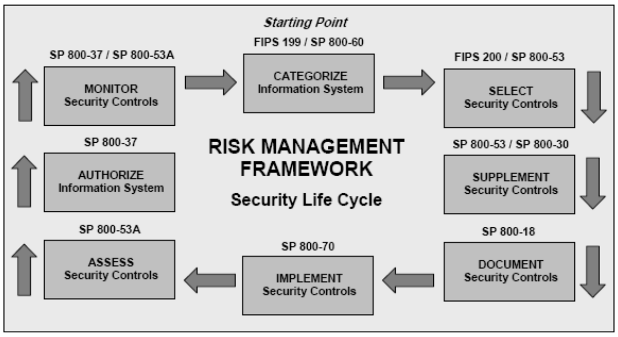


Fig. 1. Risk Management Framework (From [3])

## I. Introduction

TO ENSURE that safeguards are implemented to protect against a majority of known threats, industry leaders are requiring that information processing systems comply with specific security standards. The Federal Information Security Management Act (FISMA) enacted into law on December 17, 2002, as Title III of the E-Government Act of 2002 [1] defined three security objectives for federal government information systems: (1) Confidentiality, to preserve authorized restrictions on access and disclosure, with means for protecting personal privacy and proprietary information; (2) Integrity, to guard against improper information modification or destruction while ensuring information nonrepudiation and authenticity; and (3) Availability, to ensure timely and reliable access to and use of information [2]. To achieve these security objectives, FISMA tasked the National Institute of Standards and Technology (NIST) to develop a set of standards and guidelines, the Federal Information Risk Management Framework (RMF) (Fig. 1), that (1) describe categories for information systems according to risk levels (low, moderate, high), (2) identify types of information systems to be included in each category, and (3) describe a minimum set of security requirements (controls) that must be applied to systems in each category to achieve adequate security [4], [5], [1]. Adequate security is defined by Office of Management and Budget (OMB) Circular A-130 as “security commensurate with the risk and magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information” [6]. FISMA also requires an annual assessment of information system compliance with the required standard [7]. With approximately 100 security controls in the low-impact category to over 300 security controls in the high-impact category, the subjective compliance assessment approach described in the RMF guidance, though thorough and repeatable, lacks the clarity of a standard quantitative metric to describe for an information system the level of compliance with the standard. Given the review process outlined by NIST RMF documents, the challenge is to provide a quantitative risk analysis metric adequate to (1) clearly describe the status of compliance with the FISMA-required standard, (2) track progress toward compliance with the FISMA-required standard, (3) direct the allocation of resources required to meet FISMA minimum requirements, and (4) simplify annual report preparation. The authors propose generating a quantitative risk analysis metric at the information system level, using Pathfinder networks (PFNETs), to measure, manage, and track the status of system security compliance with the FISMA-required standard.

Approved for public release; distribution is unlimited.

## II. The RMF

### A. Purpose

The RMF, shown in Fig. 1, describes the steps and related standards and guidelines for implementing the minimum set of controls required to provide adequate security for an information system and the associated information stored, processed, and transmitted by that system. The framework includes guidance for assuring that controls are properly implemented and operating as intended to provide the expected security benefit. The RMF emphasizes the idea that risk management is a continuous process [8], [5].

### B. Federal Information Processing Standard 199

Federal Information Processing Standard (FIPS) 199 [4] addresses the first two FISMA mandates, the definition of information system categories according to risk level and the identification of system types to include in each category. FIPS 199 defines three categories for information systems considering the potential impact to organizations and individuals should a breach of confidentiality, integrity, or availability occur: (1) Low, limited adverse effect, (2) Moderate, serious adverse effect, and (3) High, severe or catastrophic adverse effect. FIPS 199 applies to all federal information systems except those designated as national security as defined in 44 United States Code Section 3542(b)(2).

### C. FIPS 200

FIPS 200 [9] addresses the third FISMA mandate, to develop minimum information security requirements (controls) for information systems in each category as defined by FIPS 199. FIPS 200 went into effect when published, March 2006. Federal agencies are required to be in compliance with the standard no later than 1 year from its effective date. There is no provision under FISMA for waivers to FIPS 200.

### D. FISMA-required System Controls

As required by FIPS 200, NIST Special Publication (SP) 800-53, Recommended Security Controls for Federal Information Systems [3], defines the security controls and provides guidelines for selecting the appropriate set to satisfy the minimum requirement for adequate security given a system category of low, moderate, or high impact. The control sets described in FIPS 200 cover 17 security-related areas (families). As illustrated in Table I, the 17 security control families are organized into three classes – management, operational, and technical – to facilitate the selection and specification of controls when evaluating an information system. Two-character identifiers are assigned to each control family. A number is appended to the family identifier to uniquely identify controls within each family. Appendix D of SP 800-53 identifies three minimum sets (baselines) of security controls that correspond to the low-, moderate-, and high-impact information system categories defined in FIPS 199. Appendix F of SP 800-53 provides a detailed description of each security control and numbered enhancements for each control where applicable. As illustrated in Table II, controls in the Access Control family not used in a particular baseline are marked Not Selected. The numbers in parentheses following the control identifiers indicate the control enhancement that applies. The baselines are intended to be broadly applicable starting points and may require modification to achieve adequate risk mitigation for a given system [3]. Given the repeatable review process outlined by the NIST RMF documents, the challenge is to provide a quantitative risk analysis metric adequate to (1) clearly describe the status of compliance with the FISMA-required standard, (2) track progress toward compliance with the FISMA-required standard, (3) direct the allocation of resources required to meet FISMA minimum requirements, and (4) simplify annual report preparation. The authors propose generating a quantitative risk analysis metric at the information system level, using PFNETs, to measure, manage, and track the status of system security compliance with the FISMA-required standard.

TABLE I
Security Control Classes, Families, and Identifiers (From [9])

| ID | FAMILY | CLASS |
|---|---|---|
| AC | Access Control | Technical |
| AT | Awareness and Training | Operational |
| AU | Audit and Accountability | Technical |
| CA | Certification, Accreditation, and Security Assessments | Management |
| CM | Configuration Management | Operational |
| CP | Contingency Planning | Operational |
| IA | Identification and Authentication | Technical |
| IR | Incident Response | Operational |
| MA | Maintenance | Operational |
| MP | Media Protection | Operational |
| PE | Physical and Environmental Protection | Operational |
| PL | Planning | Management |
| PS | Personnel Security | Operational |
| RA | Risk Assessment | Management |
| SA | System and Services Acquisition | Management |
| SC | System and Communications Protection | Technical |
| SI | System and Information Integrity | Operational |

## III. Compliance Measurement Using PFNETS

PFNETs are the result of an effort by Dearholt and Schvaneveldt [10] to develop network models for proximity data [11]. Proximity refers to the measure of relationship (similarity, relatedness, dissimilarity, distance, etc.) between two entities [10]. In networks, proximity measures are represented by distance, with small values representing similarity or a high level of relatedness, and large values representing dissimilarity or a low level of relatedness [10]. Given a dissimilarity matrix resulting from the subjective categorization (mapping) of entities as defined by Dearholt and Schvaneveldt [10], application of the Pathfinder algorithm generates a unique quantitative network representation of the proximity data. Any change in the subjective categorization of entities—in the case of risk analysis, vulnerabilities to threats—changes the resulting network. Our research indicates that the Pathfinder technique may be suitable for generating quantitative network models

TABLE II
EXCERPT FROM SECURITY CONTROL BASELINES (FROM [3])

| CNTL NO. | CONTROL NAME | CONTROL BASELINES | | |
|---|---|---|---|---|
| | | LOW | MOD | HIGH |
| AC-1 | Access Control Policy and Procedures | AC-1 | AC-1 | AC-1 |
| AC-2 | Account Management | AC-2 | AC-2(1)(2)(3)(4) | AC-2(1)(2)(3)(4) |
| AC-3 | Access Enforcement | AC-3 | AC-3(1) | AC-3(1) |
| AC-4 | Information Flow Enforcement | Not Selected | AC-4 | AC-4 |
| AC-5 | Separation of Duties | Not Selected | AC-5 | AC-5 |
| AC-6 | Least Privilege | Not Selected | AC-6 | AC-6 |
| AC-7 | Unsuccessful Login Attempts | AC-7 | AC-7 | AC-7 |
| AC-8 | System Use Notification | AC-8 | AC-8 | AC-8 |
| AC-9 | Previous Logon Notification | Not Selected | Not Selected | Not Selected |
| AC-10 | Concurrent Session Control | Not Selected | Not Selected | AC-10 |
| AC-11 | Session Lock | Not Selected | AC-11 | AC-11 |
| AC-12 | Session Termination | Not Selected | AC-12 | AC-12(1) |
| AC-13 | Supervision and Review—Access Control | AC-13 | AC-13(1) | AC-13(1) |
| AC-14 | Permitted Actions without Identification or Authentication | AC-14 | AC-14(1) | AC-14(1) |
| AC-15 | Automated Marking | Not Selected | Not Selected | AC-15 |
| AC-16 | Automated Labeling | Not Selected | Not Selected | Not Selected |
| AC-17 | Remote Access | AC-17 | AC-17(1)(2)(3)(4) | AC-17(1)(2)(3)(4) |
| AC-18 | Wireless Access Restrictions | AC-18 | AC-18(1) | AC-18(1)(2) |
| AC-19 | Access Control for Portable and Mobile Devices | Not Selected | AC-19 | AC-19 |
| AC-20 | Use of External Information Systems | AC-20 | AC-20(1) | AC-20(1) |

of information security standard controls—more accurately, the lack thereof—and information system security controls for comparison using a correlation coefficient ($cc$) formula to determine the status of information system compliance with a specified standard (%*compliant*). Among the successful applications of PFNETs have been in the discovery of salient links between documents to facilitate three-dimensional virtual reality modeling of document relationships [12], in author co-citation analysis to reveal salient linkages between groups of related authors to produce interactive author maps in real-time [13], and in the requirements phase of software development projects to determine stakeholder (users, sponsors, project managers, and developers) understanding/misunderstanding of specified requirements [14]. At a high level, the building of a PFNET involves the following steps [14]:

1) Correlate entities (e.g., vulnerabilities to threats) in an $n \times n$ matrix.
2) Build entity co-occurrence groups from entity correlations.
3) Build similarity matrix from co-occurrence groups.
4) Build dissimilarity matrix from similarity matrix.
5) Apply Pathfinder algorithm to dissimilarity matrix to build PFNET.
6) Build minimum distance matrix from PFNET.
7) Assuming steps 1 through 6 are followed to build two models of the same data entities as perceived by two different stakeholders, use a $cc$ formula to determine the degree of covariance (similarity) between the two models—quantitatively measure the similarity between two perceptions of the relationship between the same set of data entities.

As illustrated in Fig. 2, to generate the proposed %*compliant* metric, the researcher must

- Define a representative threat set where the threat level of detail is dependent on the stakeholder (e.g., system security analyst or FISMA security certifier) requirements.
- Build an *open-risk* PFNET model of the FISMA-required standard security controls. Controls when negated become vulnerabilities. Map all vulnerabilities to threat set. Complete the Pathfinder procedure.
- Build a *current-risk* PFNET model of the information system being evaluated. Map system current vulnerabilities to the threat set—mapping defined by the open-risk model (the standard). Complete the Pathfinder procedure.
- Generate current- and open-risk minimum-distance matrices from the PFNETs generated. Compare the minimum-distance matrices using a $cc$ formula to generate overall %*similar* measures for the models as well as detailed %*similar* measures for each entity within the models.
- Subtract the overall $cc$ %*similar* to open-risk measure from 1 to generate the %*compliant* to *closed-risk* (no vulnerabilities) measure.

Assuming we are evaluating a Financial Management System (FMS) that is web-enabled, intranet accessible, and categorized as moderate impact using the NIST criteria, an example using the Pathfinder technique follows.

### A. Define Representative Threat Set

Table III is a sample list of threats associated with operating the FMS application. The threat categories are taken directly or derived from Ozier [15], Bishop [16], and the Federal Information System Controls Audit Manual (FISCAM) [17].

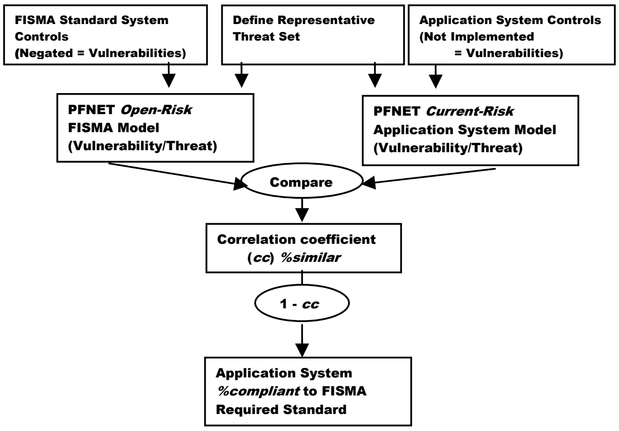

FISMA Standard System Controls (Negated = Vulnerabilities)
Define Representative Threat Set
Application System Controls (Not Implemented = Vulnerabilities)
PFNET *Open-Risk* FISMA Model (Vulnerability/Threat)
PFNET *Current-Risk* Application System Model (Vulnerability/Threat)
Compare
Correlation coefficient (*cc*) *%similar*
1 - *cc*
Application System *%compliant* to FISMA Required Standard

Fig. 2. Compliance Measurement Using Pathfinder

### B. Build open-risk PFNET Model

Table IV contains a subset of the FISMA-required baseline controls for a moderate-impact system. In Table V, the controls from Table IV are negated to create the vulnerability set for this example. To build the open-risk model (open standard) for evaluating the FMS system, we assume all 20 vulnerabilities (low-level categories) exist by mapping/relating them to the 9 threats (high-level categories) identified in Table III. Vulnerabilities may be mapped to more than one threat. This exercise may be done manually, but could become very tedious as the number of vulnerabilities and threats increases. For this example, a web-based categorization tool written by Kudikyala [18] was used to relate the vulnerabilities to threats resulting in the co-occurrence groups shown in Table VI.

The categorization tool [18] automatically builds an $n \times n$ similarity matrix of distinct entities categorized. For this example, $n$ is the sum of 9 threats and 20 vulnerabilities resulting in a $29 \times 29$ similarity matrix for the open-risk co-occurrence groups. Similarity matrix entries reflect the number of times grouped entities co-occur. For the standard open-risk co-occurrence groups, shown in Table VI, V8 and V4 co-occur 4 times. In the open-risk similarity matrix, the co-occurrence count at entries (V8, V4) and (V4, V8) would be 4.

TABLE III
THREAT CATEGORIES

| ID | Threat Category Name |
|---|---|
| T1 | Introduction of Unapproved Software |
| T2 | Software Version Implementation Errors |
| T3 | Sabotage of Software |
| T4 | Theft of Software |
| T5 | Sabotage of Data/Information |
| T6 | Theft of Data/Information/Goods |
| T7 | Destruction of Data/Information |
| T8 | Disruption of Service |
| T9 | Accountability Data Loss |

TABLE IV
FISMA STANDARD CONTROL SUBSET (FROM [3])

| ID | FISMA Control Name |
|---|---|
| AC-1 | Access Control Policy and Procedures |
| AC-2 | Account Management |
| AC-3 | Access Enforcement |
| AC-5 | Separation of Duties |
| AC-7 | Unsuccessful Login Attempts |
| AC-8 | System Use Notification |
| AC-13 | Supervision and Review–Access Control |
| AU-2 | Auditable Events [Access] |
| AU-6 | Audit Monitoring, Analysis, and Reporting |
| CM-1 | Configuration Management Policy and Procedures |
| CM-5 | Access Restrictions for Change |
| CP-4 | Contingency Plan Testing and Exercises |
| CP-9 | Information System Backup |
| CP-10 | Information System Recovery and Reconstitution |
| IA-2 | User Identification and Authentication |
| PS-4 | Personnel Termination |
| SA-5 | Information System Documentation [Operations] |
| SC-2 | Application Partitioning |
| SC-8 | Transmission Integrity |
| SI-9 | Information Input Restrictions |

TABLE V
VULNERABILITY CATEGORIES

| Control ID | Vulnerability ID | Vulnerability Category Name |
|---|---|---|
| CM-1 | V1 | Inadequate Configuration Management Policy and Procedures |
| CM-5 | V2 | Inadequate Access Restrictions for Change |
| AC-3 | V3 | Inadequate Access Enforcement |
| IA-2 | V4 | Inadequate User Identification and Authentication |
| AC-2 | V5 | Inadequate Account Management |
| AC-8 | V6 | No System Use Notification |
| AC-7 | V7 | No Termination After Maximum Unsuccessful Login Attempts |
| AC-1 | V8 | Inadequate Access Control Policy and Procedures |
| AC-13 | V9 | Inadequate Supervision and Review Access Control |
| PS-4 | V10 | Inadequate Execution of Personnel Termination Procedure |
| AU-2 | V11 | Inadequate Access Monitoring |
| SA-5 | V12 | No Information System Operations Manual |
| CP-9 | V13 | Insufficient System Backups |
| CP-10 | V14 | Inadequate Recovery Mechanisms |
| CP-4 | V15 | No Contingency Plan Testing and Exercises |
| AU-6 | V16 | Inadequate Audit Monitoring, Analysis, and Reporting |
| SC-8 | V17 | Integrity of Transmitted Data not Protected |
| AC-5 | V18 | Inadequate Separation of Duties |
| SC-2 | V19 | Inadequate Application Partitioning |
| SI-9 | V20 | Inadequate Information Input Restrictions |

TABLE VI
STANDARD OPEN-RISK CO-OCCURRENCE GROUPS

| |
|---|
| (T1, V1) |
| (T2, V1) |
| (T3, V2) |
| (T4, V2) |
| (T5, V18, V17, V10, V8, V4, V3, V1) |
| (T6, V18, V10, V8, V4, V3, V1) |
| (T7, V20, V19, V10, V8, V4, V3, V1) |
| (T8, V15, V14, V13, V12, V3, V1) |
| (T9, V16, V11, V9, V8, V7, V6, V5, V4) |

Higher co-occurrence counts indicate greater similarity. The categorization tool [18] automatically builds a dissimilarity matrix from the similarity matrix of categorized entities. The vulnerability-to-threat relationships in this example are symmetric. Therefore an open-risk dissimilarity matrix (upper triangular portion only) is generated from the open-risk similarity matrix by subtracting each co-occurrence count entry from the maximum co-occurrence count entry plus one to prevent 0-value dissimilarity matrix entries. Lower co-occurrence counts indicate greater similarity.

A Unix-based PFNET generation tool, written by Kurup [19] applying the Dearholt and Schvaneveldt algorithm, was used to generate the open-risk PFNET from the dissimilarity matrix. The tool requires as input the number of nodes ($n = 29$), the upper triangular portion of the dissimilarity matrix, an $r$-metric ($\infty$, input as $-1$), and a $q$ parameter ($n - 1 = 28$). A path will exist between node pair $(i, j)$ in PFNET $(r, q)$ if and only if there is no shorter alternate path between $(i, j)$, where $r$ is the Minkowski $r$-metric calculation of path weight, for paths with number of links $\leq q$.

The distance between two nodes not directly linked is computed using the Minkowski $r$-metric. For path $P$ with weights $w_1, w_2, \ldots, w_k$, the Minkowski distance is [10], [14]

$$w(P) = \left( \sum_{i=1}^{k} w_i^r \right)^{1/r} \qquad \text{where } r \geq 1, w_i \geq 0 \text{ for all } i. \quad (1)$$

When $r = 1$, path weight is calculated by summing the link weights along the path [10], [14]. Calculating path weight this way assumes ratio-scale data where each weight value is presumed to be within a multiplicative constant of the "correct" value [10]. When link values are obtained from empirical data, computing path weight this way may not be justifiable [20]. For generating PFNETs, where only the ordinal relationships between link weights and path weights are important, $r$ should be set to $\infty$ [10]. When $r = \infty$, the path weight is the same as the maximum weight associated with any link along the path [10], [14].

The PFNET generated from the open-risk dissimilarity matrix is a mathematical model of standard open risk.

### C. Build current-risk PFNET Model

Assume these vulnerabilities exist in the FMS system: V4, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, and V20 (see Table V). To build the current-risk PFNET model, map the FMS vulnerabilities to threats as dictated by the vulnerability mappings in the open-risk standard model to generate the co-occurrence groups shown in Table VII under "FMS System Current Risk." (Note: the Standard Open Risk and FMS System Current Risk co-occurrence groups in Table VII are the initial entries in Table VIII.)

TABLE VII
CO-OCCURRENCE GROUPS

| Standard Open Risk | FMS System Current Risk |
|---|---|
| (T1, V1) | |
| (T2, V1) | |
| (T3, V2) | |
| (T4, V2) | |
| (T5, V18, V17, V10, V8, V4, V3, V1) | (T5, V18, V17, V10, V8, V4) |
| (T6, V18, V10, V8, V4, V3, V1) | (T6, V18, V10, V8, V4) |
| (T7, V20, V19, V10, V8, V4, V3, V1) | (T7, V20, V19, V10, V8, V4) |
| (T8, V15, V14, V13, V12, V3, V1) | (T8, V15, V14, V13, V12) |
| (T9, V16, V11, V9, V8, V7, V6, V5, V4) | (T9, V16, V11, V9, V8, V7, V6, V4) |

Using the procedure described in Section III-B

- A similarity matrix is generated from the FMS system current-risk co-occurrence groups.
- A dissimilarity matrix is generated from the similarity matrix.
- The PFNET algorithm is applied to the dissimilarity matrix to generate the current-risk PFNET model.

The PFNET generated from the current-risk dissimilarity matrix is a mathematical model of the FMS system current risk.

### D. Compare Minimum Distance Matrices

A Unix-based PFNET correlation tool, written by Kudikyala [21], was used to generate minimum distance matrices from the standard open-risk and FMS system current-risk PFNETs using Floyd's algorithm for shortest path [22]. Path distances for the minimum distance matrices are calculated the traditional way, by adding link weights along paths between nodes. The correlation tool was also used to compare the open- and current-risk minimum distance matrices using the $cc$ formula that follows:

$$cc = \frac{\sum (a - \bar{a})(b - \bar{b})}{\sqrt{\sum (a - \bar{a})^2 \sum (b - \bar{b})^2}} \qquad (2)$$

where $a$ is the value of an element in the distance vector of the open-risk minimum distance matrix, $\bar{a}$ is the mean of all the elements in the open-risk distance vector (upper or lower triangular values), $b$ is the value of a corresponding element in the distance vector of the system current-risk minimum distance matrix, and $\bar{b}$ is the mean of all elements in the current-risk distance vector. Normally the $cc$ range is $[-1, +1]$, where $-1$ represents no similarity and $+1$ represents perfect

TABLE VIII
RISK MODEL CO-OCCURRENCE GROUPS

| | | |
|---|---|---|
| **Open Risk:** | (T1,**V1**) | (T2, **V1**) |
| See Table VII | (T3,**V2**) | (T4,**V2**) |
| | (T5,V18,V17,V10,V8,V4,**V3**,**V1**) | (T6,V18,V10,V8,V4,**V3**,**V1**) |
| | (T7,V20,V19,V10,V8,V4,**V3**,**V1**) | (T8,V15,V14,V13,V12,**V3**,**V1**) |
| | (T9,V16,V11,V9,V8,V7,V6,**V5**,V4) | |
| **FMS Model 1** | (T5,V18,**V17**,V10,V8,**V4**) | (T6,V18,V10,V8,**V4**) |
| See Table VII | (T7,V20,V19,V10,V8,**V4**) | (T8,V15,V14,V13,V12) |
| | (T9,V16,V11,V9,V8,V7,V6,**V4**) | |
| **FMS Model 2** | (T5,**V18**,V10,V8) | (T6,**V18**,V10,V8) |
| | (T7,**V20**,**V19**,V10,V8) | (T8,V15,V14,V13,V12) |
| | (T9,V16,V11,V9,V8,V7,V6) | |
| **FMS Model 3** | (T5,**V10**,V8) | (T6,**V10**,V8) |
| | (T7,**V10**,V8) | (T8,V15,V14,V13,V12) |
| | (T9,V16,V11,V9,V8,V7,V6) | |
| **FMS Model 4** | (T5,V8) | (T6,V8) |
| | (T7,V8) | (T8,V15,V14,V13,V12) |
| | (T9,**V16**,**V11**,V9,V8,**V7**,V6) | |
| **FMS Model 5** | (T5,V8) | (T6,V8) |
| | (T7,V8) | (T8,V15,V14,V13,**V12**) |
| | (T9,**V9**,V8,V6) | |
| **FMS Model 6** | (T5,V8) | (T6,V8) |
| | (T7,V8) | (T8,**V15**,**V14**,**V13**) |
| | (T9,V8,V6) | |
| **FMS Model 7** | (T5,**V8**) | (T6,**V8**) |
| | (T7,**V8**) | (T9,**V8**,V6) |
| **FMS Model 8** | (T9,**V6**) | |
| **Closed Risk** | (No Vulnerabilities) | |
| Note: Vulnerabilities in bold type assumed corrected in following model | | |

similarity between models [14], [23]. Because of the approach taken in this research to compare current system state to a standard perception of adequate security, the $cc$ range is narrowed from $[-1, +1]$ to $[0, +1]$—no comparison beyond a perfect match.

### *E. Generate %*compliant *Measure*

The correlation tool [21] generates an overall $cc$ value that indicates the degree of covariance (similarity) between the standard open-risk model and the system current-risk model – similarity to unacceptable risk; all vulnerabilities exist. The goal for the FMS system is a $cc$ of $0$, i.e., no similarity to the open-risk model. Subtracting the overall $cc$ value from $1$ yields a value (%*compliant*) that indicates how close the FMS system is to standard compliance as defined by the closed-risk model—no vulnerabilities exist. Comparing the FMS system current-risk model 1 to the open-risk model results in a $cc$ of $0.45$ (see Table IX,“Overall Path Distance $cc$” for FMS 1). The FMS 1 current-risk model in this example exhibits $45$ percent similarity to the open-risk model. Subtracting $0.45$ from $1.0$ (open-risk) yields a value that indicates the FMS system is $55$ percent compliant to closed-risk (see Table IX “%*compliant*” for FMS 1). The more existing vulnerabilities identified in the FMS system, the closer the resulting $cc$ value will be to $1.0$ (open-risk). As vulnerabilities are removed, the $cc$ value moves closer to $0.0$ (closed-risk). Table VIII shows sample FMS risk model co-occurrence groups. The vulnerabilities in bold type are removed in each successive FMS model. For each FMS model, the Pathfinder procedure was applied to generate a minimum distance matrix for comparison with the open-risk model minimum distance matrix. Table IX shows the overall path distance $cc$, node path distance (detailed) $cc$, and %*compliant* values for the FMS models as vulnerabilities are removed and the FMS models are compared with the open-risk model.

Using the distance vectors for each entity in the minimum distance matrices for the open- and current-risk models, detailed $cc$ values are generated that indicate how a single entity in each model relates to all others—how a single entity contributes to the similarity between models. An analysis of the detailed $cc$ values for models compared should provide some insight with regard to choosing an efficient mitigation path to reaching compliance with standard.

## IV. CONCLUSION

Technical Topic Area 3 (TTA 3), Cyber Security Metrics, of the Department of Homeland Security Broad Agency Announcement (BAA), Cyber Security Research and Development (BAA07-09) [24], describes security metrics as “a difficult, long-standing problem.” TTA 3 cites the fact that the security metrics problem is listed on the INFOSEC Research Council (IRC) Hard Problems List [25] as evidence of the importance of research in this area. Good security metrics are required to direct the allocation of security resources to improve the security status of government information systems, to demonstrate compliance with FISMA-required

TABLE IX
RISK MODEL COMPARISONS

| | **Open Risk** | **FMS 1** | **FMS 2** | **FMS 3** | **FMS 4** | **FMS 5** | **FMS 6** | **FMS 7** | **FMS 8** | **Closed Risk** |
|---|---|---|---|---|---|---|---|---|---|---|
| **%***compliant* | 0.00 | 0.55 | 0.59 | 0.66 | 0.69 | 0.77 | 0.82 | 0.87 | 0.95 | 1.00 |
| **Overall Path Distance** ***cc*** | 1.00 | 0.45 | 0.41 | 0.34 | 0.31 | 0.23 | 0.18 | 0.13 | 0.05 | 0.00 |
| **Node Path Distance** ***cc*** | | | | | | | | | | |
| V1 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V2 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V3 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V4 | 1.00 | 0.70 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V5 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V6 | 1.00 | 0.43 | 0.46 | 0.45 | 0.62 | 0.41 | 0.33 | 0.33 | 0.23 | 0.00 |
| V7 | 1.00 | 0.43 | 0.46 | 0.45 | 0.62 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V8 | 1.00 | 0.70 | 0.43 | 0.35 | 0.13 | 0.10 | 0.09 | 0.09 | 0.00 | 0.00 |
| V9 | 1.00 | 0.43 | 0.46 | 0.45 | 0.63 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 |
| V10 | 1.00 | 0.75 | 0.54 | 0.43 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V11 | 1.00 | 0.43 | 0.46 | 0.45 | 0.63 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V12 | 1.00 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.00 | 0.00 | 0.00 | 0.00 |
| V13 | 1.00 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.48 | 0.00 | 0.00 | 0.00 |
| V14 | 1.00 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.48 | 0.00 | 0.00 | 0.00 |
| V15 | 1.00 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.48 | 0.00 | 0.00 | 0.00 |
| V16 | 1.00 | 0.43 | 0.46 | 0.45 | 0.62 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V17 | 1.00 | 0.66 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V18 | 1.00 | 0.74 | 0.69 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V19 | 1.00 | 0.65 | 0.62 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| V20 | 1.00 | 0.65 | 0.62 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| T1 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| T2 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| T3 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| T4 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| T5 | 1.00 | 0.66 | 0.64 | 0.54 | 0.29 | 0.29 | 0.29 | 0.29 | 0.00 | 0.00 |
| T6 | 1.00 | 0.64 | 0.61 | 0.51 | 0.31 | 0.31 | 0.31 | 0.31 | 0.00 | 0.00 |
| T7 | 1.00 | 0.64 | 0.62 | 0.55 | 0.29 | 0.29 | 0.29 | 0.29 | 0.00 | 0.00 |
| T8 | 1.00 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.48 | 0.00 | 0.00 | 0.00 |
| T9 | 1.00 | 0.43 | 0.46 | 0.45 | 0.62 | 0.41 | 0.33 | 0.33 | 0.23 | 0.00 |

security control standards, and to simplify the annual FISMA reporting requirement. TTA 3 advises that "the lack of sound and practical security metrics is severely hampering progress both in research and engineering of secure systems" [24].

The proposed approach is unique in that it offers a %*compliant* metric at the information system level. The proposed approach in combination with NIST RMF guidance provides for producing consistent quantitative results. Detailed $cc$ values should indicate vulnerability groups where targeted cost benefit analysis may be applied to determine an effective approach for eliminating vulnerabilities contributing most to the noncompliant state of the system being evaluated. The quantitative %*compliant* metric should allow for the discussion of system compliance with FISMA-required standards in terms easily understood by participants at various levels of an organization without requiring all to have detailed knowledge of the internals of the security standard or the system being evaluated.

## ACKNOWLEDGEMENT

E. Hulitt and Dr. R.B. Vaughn thank Marsha Gay for her expert editing of this article.

## REFERENCES

[1] United States General Accounting Office, "Information security: Agencies need to implement consistent processes in authorizing systems for operations," Report to Congressional Requesters, (GAO-04-376), http://www.gao.gov/cgi-bin/getrpt?GAO-04-376, Tech. Rep., 2004.

[2] United States Public Law 107-347-DEC. 17 2002, 116 STAT. 2899, *Federal Information Security Management Act (FISMA)*. Title III of the E-Government Act of 2002, http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ347.107.pdf, 2002.

[3] National Institute of Standards and Technology, *Recommended Security Controls for Federal Information Systems, SP 800-53 Rev. 1*. Gaithersburg, MD USA: Computer Security Division, http://csrc.nist.gov/publications/nistpubs/800-53-Rev1/800-53-rev1-final-clean-sz.pdf, 2006.

[4] ——, *Standards for Security Categorization of Information Systems, FIPS PUB 199*. Gaithersburg, MD: Computer Security Division, http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf, 2004.

[5] R. Ross, S. Katzke, and P. Toth, "The new FISMA standards and guidelines changing the dynamic of information security for the federal government," in *2005 IEEE Military Communications Conference*, vol. 2. Atlantic City, NJ: IEEE, October 17-21 2005, pp. 864–870.

[6] United States Office of Management and Budget (OMB), *Security of federal automated information resources, Appendix III to OMB Circular No. A-130*. Management of Federal Information Resources, http://www.whitehouse.gov/omb/circulars/a130/a130.html, February 1996.

[7] National Institute of Standards and Technology, *Security Metrics Guide for Information Technology Systems, SP 800-55*. Gaithersburg, MD: Computer Security Division, http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf, 2003.

[8] M. Gerber and R. von Solms, "Management of risk in the information age," *Computers & Security*, vol. 24, no. 1, pp. 16–30, February 2005.

[9] National Institute of Standards and Technology, *Minimum Security Requirements for Federal Information and Information System, FIPS PUB 200*. Gaithersburg, MD: Computer Security Division, http://csrc.nist.gov/publications/fips/fips200/FIPS-PUB-200-final-march.pdf, 2006.

[10] D. Dearholt and R. Schvaneveldt, "Properties of pathfinder networks," in *Pathfinder Associative Networks: Studies in Knowledge Organization*, R. Schvaneveldt, Ed. Norwood, NJ: Ablex Publishing Corporation, 1990, pp. 1–30.

[11] R. Schvaneveldt, "Preface," in *Pathfinder Associative Networks: Studies in Knowledge Organization*, R. Schvaneveldt, Ed. Norwood, NJ: Ablex Publishing Corporation, 1990, p. ix.

[12] C. Chen, "Bridging the gap: The use of pathfinder networks in visual navigation," *Journal of Visual Languages and Computing*, vol. 9, no. 3, pp. 267–286, 1998.

[13] X. Lin, J. Buzydlowski, and H. White, "Real-time author co-citation mapping for online searching," *Information Processing and Management*, vol. 39, no. 5, pp. 689–706, 2003.

[14] U. Kudikyala, "Reducing misunderstanding of software requirements by conceptualization of mental models using pathfinder networks," Ph.D. dissertation, Department of Computer Science, Mississippi State University, Starkville, MS, 2004.

[15] W. Ozier, "Risk analysis and assessment," in *Information Security Management Handbook*, 5th ed., H. Tipton and M. Krause, Eds. Boca Raton, FL: Auerbach Publications, 2004, pp. 795–820.

[16] M. Bishop, *Computer Security: Art and Science*. Boston, MA: Addison-Wesley, 2003.

[17] United States General Accounting Office, *Federal Information System Controls Audit Manual (FISCAM), Volume I Financial Statement Audits, (GAO/AIMD-12.19.6)*. Accounting and Information Management Division, 1999.

[18] U. Kudikyala, "Requirements categorization tool," Department of Computer Science, Mississippi State University, Starkville, MS, Tech. Rep., February 2003.

[19] G. Kurup, "PFNET generation tool (geom_pfn)," Department of Computer Science, Mississippi State University, Starkville, MS, Tech. Rep., August 1989.

[20] R. Schvaneveldt, "Graph theory and Pathfinder primer," in *Pathfinder Associative Networks: Studies in Knowledge Organization*, R. Schvaneveldt, Ed. Norwood, NJ: Ablex Publishing Corporation, 1990, pp. 297–299.

[21] U. Kudikyala, "Pfnet comparison tool (correlations.java)," Department of Computer Science, Mississippi State University, Starkville, MS, Tech. Rep., February 2003.

[22] T. Corman, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.

[23] U. Kudikyala and R. B. Vaughn, "Understanding software requirements using Pathfinder networks," *CrossTalk: The Journal of Defense Software Engineering*, vol. 17, no. 5, pp. 16–25, May 2004.

[24] United States Department of Homeland Security, *Cyber Security Research and Development*. Broad Agency Announcement BAA07-09, http://www.hsarpabaa.com/Solicitations/BAA07-09_CyberSecurityRD_Posted_05162007.pdf, 2007.

[25] INFOSEC Research Council (IRC), "National scale INFOSEC research hard problems list, draft 21," http://www.infosec-research.org/documents, September 1999.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 807 – 811
ISBN 978-83-60810-14-9
ISSN 1896-7094

# Analysis of Different Architectures of Neural Networks for Application in Intrusion Detection Systems

Przemysław Kukiełka
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
Email: Przemyslaw.Kukielka@telekomunikacja.pl

Zbigniew Kotulski
Institute of Fundamental Technological Research
Polish Academy of Sciences
Swietokrzyska 21, 00-049 Warsaw, Poland
Email: zkotulsk@ippt.gov.pl

***Abstract*—Usually, Intrusion Detection Systems (IDS) work using two methods of identification of attacks: by signatures that are specific defined elements of the network traffic possible to identification and by anomalies being some deviations form of the network behavior assumed as normal. In the both cases one must pre-define the form of the signature (in the first case) and the network's normal behavior (in the second one). In this paper we propose application of Neural Networks (NN) as a tool for application in IDS. Such a method makes possible utilization of the NN learning property to discover new attacks, so (after the training phase) we need not deliver attacks' definitions to the IDS. In the paper, we study usability of several NN architectures to find the most suitable for the IDS application purposes.**

## I.INTRODUCTION

BECAUSE of their generalization feature, neural networks are able to work with imprecise and incomplete data. It means that they can recognize also patterns not presented during a learning phase. That is why the neural networks could be a good solution for detection of a well-known attack, which has been modified by an aggressor in order to pass through the firewall system. In that case, traditional Intrusion Detection Systems, based on the signatures of attacks or expert rules, may not be able to detect the new version of this attack.

In this paper, we focus on three different network architectures: Backpropagation, Radial Basis Function and Self Organizing Map. The result of simulation is the information about the attack detection accuracy, represented as a number of false attacks and not detected attacks in comparison to number of validation vectors for each type of used NN. Performed tests allow us to find drawback of usage of each type of architectures for detection a specific type of the attack. As the input vector, we used data produced by the KDD99 project.

## II.NEURAL NETWORK: A WAY OF WORK

An artificial neural network is a system simulating a work of the neurons in the human brain. In Fig. 1 it is presented the diagram of a neuron's operation.

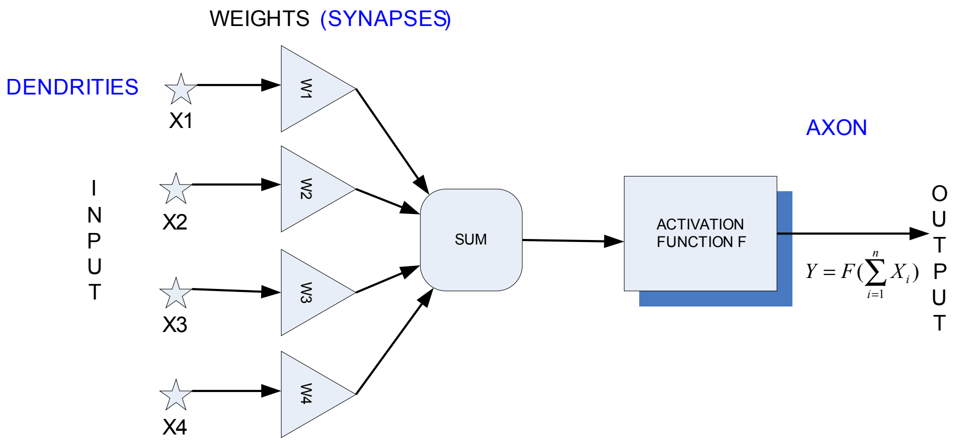


Fig. 1. A scheme of an artificial neuron

This is the work in progress

The neuron consists of some inputs emulating dendrites of the biological neuron, a summation module, an activation function and one output emulating an axon of the biological neuron. The importance of a particular input can be intensified by the weights that simulate biological neuron's synapses. Then, the input signals are multiplied by the values of weights and next the results are added in the summation block. The sum is sent to the activation block where is pro cessed by the activation function. Thus, we obtained neuron's answer for the input signals "x".

### *A. MLP (Multi Layer Perceptron)*

One neuron cannot solve a complex problem that is why the neural network consisted of many neurons is used. One of the most often used architecture is the Multi Layer Perceptron. In such a network, all neurons' outputs of the previous layer are connected with neurons' inputs of the next layer. The MLP architecture consists of one or more hidden layers. A signal is transmitted in the one direction from the input to the output and therefore this architecture is called feedforward. The MLP networks are learned with using the Backward Propagation algorithm (BP). In order to reach better efficient and speed of learning process it arise many types of BP algorithm. In our research we used following variants of the BP algorithm:

- Batch Gradient descent (traingd)
- Batch Gradient descent with momentum (traingdm)
- Levenberg-Marquardt (trainln)
- Resilient Backpropagation (trainrp)
- Conjugate Gradient (traincgf)
- Quasi Newton (trainbfg)
- Quasi Newton 2 (trainoss)

For the simulation procedure, the Matlab toolbox was used. The variants of BP algorithm are followed by the names of the learning Matlab's functions in the bracket.

### *B. Radial Based Function (RBF) Neural Network*

The radial neuron networks in comparison to the MLP where a global approximation is used are working based on the local approximation.

Typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function φ and a linear output layer.

Activation function for the radial neuron network is:

$$\varphi(x) = \exp\left(\frac{\|x - c_i\|^2}{2\sigma_i^2}\right),$$

where C- Center of the function $\sigma$ Spread parameter of the radial function

The argument of radial function φ is the Euclidean distance sample x from center c.

In the RBF network, there are three types of parameters that need to be chosen to adapt the network for a particular task: the center vectors $c_i$ , the output weights $w_i$ , and the RBF spread parameter $\beta_i$.

### *C. SOM (Self Organizing Maps)*

This kind of network is learned without teacher based on the competition rules. On the input of such network learning vector is put and in the next step distance from it to the weight vector is checked. Neuron for its distance is lowest becomes the winner and can modify values of his weights. Depends of the SOM networks architecture the weighs of the winner neuron can be modified (Winner Takes all) or weigh of the winner and neurons from his neighbor (Winner Takes most).

More information about neural network could be found in [2], [7].

## III. KDD99 Input Data

We are using the KDD 99 data set as the input vectors for training and validation of the tested neural network. It was created based on the DARPA (Defense Advanced Research Project Agency) intrusion detection evaluation program. MIT Lincoln Lab that participates in this program has set up simulation of typical LAN network in order to acquire raw TCP dump data [1]. They simulated LAN operated as an normal environment, which was infected by various types of attacks. The raw data set was processed into connection records. For each connection, 41 various features were extracted. Each connection was labeled as normal or under specific type of attack. Four main categories of attacks were simulated:

- **DoS** (Denial of Service): an attacker tries to prevent legitimate users from using a service e.g. TCP SYN Flood, Smurf .
- **Probe:** an attacker tries to find information about the target host. For example: scanning victims in order to get knowledge about available services, using Operating System etc.
- **U2R** (User to Root): an attacker has local account on victim's host and tries to gain the root privileges.
- **R2L** (Remote to Local): an attacker does not have local account on the victim host and try to obtain it.

The KDD data sets are divided into tree subsets: 10%KDD, corrected KDD, whole KDD. Basis characteristic of KDD data sets are shown in Table I. It includes number of

Table I.
KDD99 data subsets

| Dataset | DoS | Probe | U2r | U2l | Normal |
|---|---|---|---|---|---|
| 10%KDD | 391458 | 4107 | 52 | 1126 | 97277 |
| Corrected KDD | 229853 | 4166 | 70 | 16347 | 60593 |
| Whole KDD | 3883370 | 41102 | 52 | 1126 | 972780 |

TABLE II.
DATA SUBSETS FOR TRAINING PROCESS

| Data Set name for training process | Number of normal connection | Number of connection labelled as attack . |
|---|---|---|
| Learn_set | 1000 | 8653 |
| Learn_set_radial_trad | 100 | 1179 |
| Learn_set_radial | 1000 | 1179 |

connections assigned to the particular class (DoS, Probe etc.)..

### A. 10%KDD

The 10%KDD data set is used for the training process of the IDS. It includes connections simulated following 22 types of the attacks: back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop. The attacks are not represented by the same number of connections. The most of the simulated attacks are of DoS group because of the nature of this type attacks that are using many IP packets in order to block network services.

### B. Corrected KDD

The Corrected KDD data set is used for testing process of IDS. It includes additional 14 types of new attack not presented in 10%KDD and Whole KDD. Thanks to them it is possible to check if tested IDS is able to detect new attack not presented in the training phase.

In our research only 10 %KDD and corrected KDD data set was used.

## IV. TRAINING PROCESS

### A. Process of Selection of the Input Vector

The data set 10% KDD includes the large number of connection. It is influence on the long time of training, high requirements for efficient of using implementation of neural network and hardware on that it is working. That's why in research presented in this publication for training purpose three randomly selected small date set was used. Table II shows how many connections are assigned to the particular training data set.

It was chosen the same number of connections represent each type of attack. In case when number of connections for particular type of attack was lower than assumed number all connections for this group was selected. Because some features of connection (e.g. protocol, flags) were existed as a characters string, it was transformed to numerical representation.

Moreover, for the SOM neural network normalization process of the input data "xi" was performed.

$$x_i^{'} = \frac{x_i}{\sqrt{\sum_{i=1}^{n} x_i^2}} \ .$$

For Radial Neuron network it was noticed that the best results are obtained when the value of features is decreased by dividing it by constant number equal 1000.

For the research purpose three type of neural network architectures were used: MLP (multilayer perceptron), RBF (Radial Basis Function), SOM (Self Organizing Maps).

***Main Assumption for Training Process of MLP***

Number of Epochs = 1000.
MSE (Mean Square error) = 0.01.
Learning rate =1.
Momentum = 0.6.
Activation function =log-signoid.
Number of neurons in hidden layer=5.
Number of neurons in output layer =1.
Updates of weighs – batch mode (after presentation of entire training data set).

Results of the training process show that only the algorithms: **Levenberg-Marquardt and Quasi Newton** achieved to fulfill the above assumption. In other case MLP network cannot reach assumed MSE =0.1 in 1000 Epochs of training process. That is why in the next simulation we focus only on networks that can be trained according our assumptions.

### B. For RBF Network

Implementation from Matlab toolbox was used (newrb**).** At the beginning, a network without radial neurons in the first layer is created. In the next step, the MSE is calculated and a new radial neuron with weighs equals to input vector that caused the max value of MSE is added to the first layer. The last operation is modification of weights of a lineal neuron in the output layer in direction of minimize MSE. All steps are repeated until the assumed MSE is reached.

***Main Assumption for Training Process***

MSE=0.1(because of high calculation power requirements not possible to achieve MSE =0.01 like for MLP)
Spread parameter $\sigma$ = 3
MN (Maximum number of neurons) = 250
DF (Number of neurons to add between displays) = 2
MNE (Number of neurons for that RBF achieved MSE =0.1) = 178
Number of Epochs = 178

### C. For SOM Network

***Main Assumption for Training Process***

Implementation of SOM in Matlab environment was used (function newsom).
Number of Epochs = 1000.
Neighbors topology= hextop
Distance function – mean as a number of links between neurons or steps that must be taken to get neuron under consideration = matlab linkdist.
Ordering Phase learning rate =0.9.

TABLE III.
THE DATA SUBSET FOR TESTING PHASE

| Data Set name for testing process | Number of normal connection | Number of connection labelled as attack . |
|---|---|---|
| Test_set | 60593 | 250436 |
| Test_set_noanomaly | 60593 | 231694 |

TABLE IV.
COMPARISON OF DIFFERENT NETWORK TOPOLOGY TRAINED WITH "LEARN_SET_RADIAL_TRAD" DATA SET

| Neural Network topology | False alarm Number/ Percent of all normal connection | Not detected/ Detection rate | Duration of training |
|---|---|---|---|
| **MLP** learning algorithm<br>**Levenberg-Marquardt** | 31594<br>52% | 382<br>98,5% | About 4 seconds |
| **MLP** learning algorithm<br>**Quasi Newton** | 37530<br>61% | 90<br>99% | About 4 seconds |
| **Radial** | 21938<br>36% | 61568<br>76% | About 2 minutes 20 seconds |
| **SOM** | 19677<br>32% | 40957<br>84% | About 7 minutes |

TABLE V.
RESULTS OF SIMULATION OF MLP NETWORK LEARNED WITH USAGE INPUT DATE SET „LEARN_SET" AND „LEARN_SET_RADIAL"

| Variations of BP algorithm used for training MLP Network | False alarm Number/ Percent of all normal connection | Detection rate |
|---|---|---|
| **MLP** learning algorithm Levenberg-Marquardt<br>**Input data set:** Learn_set, Test_set | 5354<br>8,8% | 19207<br>92% |
| **MLP** learning algorithm Levenberg-Marquardt<br>**Input data set:** Learn_set, Test_noanomaly | 5351<br>8,9% | 4097<br>98% |
| **MLP** learning algorithm Levenberg-Marquardt<br>**Input data set:** Learn_set_radial, Test_set | 10068<br>16,6% | 17139<br>93% |
| **MLP** learning algorithm Quasi Newton<br>**Input data set:** Learn_set, Test_set | 4790<br>7,9% | 10294<br>94% |
| **MLP** learning algorithm Quasi Newton<br>**Input data set:** Learn_set, Test_set_noanomaly | 4790<br>7,9% | 87<br>99,9% |
| **MLP** learning algorithm Quasi Newton<br>**Input data set:** Learn_set_radial, Test_set | 31894<br>52% | 1556<br>99% |
| **MLP** learning algorithm Resilient Backpropagation<br>**Input data set:** Learn_set, Test_set | 4453<br>7,3% | 12877<br>95% |
| **MLP** learning algorithm Resilient Backpropagation<br>**Input data set:** Learn_set, Test_set_noanomaly | 4454<br>7,35% | 2714<br>99% |
| **MLP** learning algorithm Resilient Backpropagation<br>**Input data set:** Learn_set_radial, Test_set | 3564<br>5,9% | 29805<br>88% |

Ordering Phase steps = 1000.
Tuning phase learning rate =0.02.
Tuning phase neighbor distance = 1.

In the ordering phase neighbor distance between two neurons decreases from maximum values to the Tuning phase values, the learning rate decreases from the Ordering phase learning rate to the Tuning phase learning rate. Neuron's weights are expected to order themselves in the input space consistent with the associated neurons position.

In the Tuning phase neighbor distance stay on the same level equal Tuning phase neighbor distance, the learning rate continues to decrease but very slowly. Neuron's weights are expected to spread out ever the input space relatively evenly while retaining their topological order obtained during the ordering phase..

## V. RESULTS OF THE TESTS

The neural networks architectures were tested with using whole date set from the Corrected KDD and with using modified data set that includes only attacks presented during training process. Data sets used for the testing phase were shown in Table III.

In Table IV is presented the comparison of particular neural network architecture learned with input data from „Learn_radial_trad" data set. The evaluation focuses on the value of the detection rate and the false alarm rate for "Test_set" data set. For analysis of neural network answer, it was assumption that value from 0 to 0.5 concerning "normal" and from 0.5 to 1 "attack". The Corrected Data Set used for testing phase includes 311 029 vectors.

High number of false alarms is a result of small number of "normal" connections in the trained data set. It was reduced because of a problem of efficiency of SOM and RBF NN implementation in the Matlab environment.

The MLP network was the most efficient during comparison of different network architecture. That is why in the further research we plan to focus on it. For the training of the MLP network we used an input data set with bigger number of connection (Learn_set, Learn_set_radial). The experiment should show if increasing the number of input patterns improves detection rate and false alarms rate. The results of this test are presented in Table V.

## VI. Conclusions

Usage of neural networks for intrusion detection with the input data from DARPA project was presented in many publication. Unfortunately, in description of simulation process very often is lack of information about assumptions that were made in before. For instance there is no information if the whole tests data set was used in the simulation or only some its subset.

The goal of this research was the comparison of different neural network architectures working with the same assumed parameters and tested with with usage of the whole DARPA tests data set (Corrected KDD). Our research made it possible to formulate precisely general assumptions for making benchmark simulations as well as gave us some conclusions concerning particular NN architectures applied to IDS. The main conclusions are:

- Selection of the input data is a very important issue. Representation of all types of attacks and normal activity should be included in the learning date set.
- Results of simulation show that detection rate was the best for MLP network learned with Backward Propagation Levenberg-Marquardt algorithm
- In the second phase of simulation MLP network number of input vectors are increased because of that summary amount of errors decreases. In particular number of false alarms decreases and numbers of non detected attack a little increases. The reason is that representation of more different type of normal activity was added to input vectors in learning phase.
- The long time is required for the learning phase of SOM and Redial neural network. Moreover, these two neural network architectures need very high CPU performance and Operational Memory (RAM). During tests phase input date set had to be divided into smaller subsets in order to avoid "lack of memory swap" Matlab error.
- MLP required less CPU performance and Operational Memory (RAM). That is why could be tested with using the whole tests data set not divided into smaller subsets.

## References

[1] W. Lee, S. J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems", *ACM Transactions on Information and System Security* (TISSEC), 3(4): 227-261, 2000.

[2] L. Rutkowski, *Metody i techniki sztucznej inteligencji* , PWN, Warszawa 2005. (In Polish)

[3] W. Lee, S. J. Stolfo, "Data Mining Approaches for Intrusion Detection", *Proceedings of the Seventh USENIX security Symposium* (SECURITY '98), San Antonio, TX,1998.

[4] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, K. Das, "The 1999 Darpa Off-Line Intrusion Detection Evaluation", *Computer Networks: The International Journal of Computer and Telecommunications Networking* 34 (2000) 579-595, 2000.

[5] V. Paxson," Bro: A system for Detecting Network Intruder in Real Time" In *Proceedings of the 7th USENiX Security Symposium*, San Antonio 1998.

[6] Ch. Elkan, "Results of the KDD'99 Classifier-learning contest", In `http://www-cse.ucsd.edu/#elkan/clresults.html', September 1999.

[7] S. Osowski, *Sieci neuronowe do przetwarzania informacji* , Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000, ISBN 83-7207-187-X. (In Polish)

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Semantic Framework for Privacy-Aware Access Control

Georgios V. Lioudakis, Nikolaos L. Dellas, Eleftherios A. Koutsoloukas,
Georgia M. Kapitsaki, Dimitra I. Kaklamani, Iakovos S. Venieris
National Technical University of Athens, Heroon Polytechniou 9, 15773, Athens, Greece
Email: {gelioud, ndellas, lefterisk, gkapi}@icbnet.ntua.gr, dkaklam@mail.ntua.gr, venieris@cs.ntua.gr

***Abstract*—The issue of privacy is constantly brought to the spotlight since an ever increasing number of services collects and processes personal information from users. In fact, recent advances in mobile communications, location and sensing technologies and data processing are boosting the deployment of context-aware personalized services and the creation of smart environments but, at the same time, they pose a serious risk on individuals' privacy rights. Being situated in the realms of legal and social studies, the notion of privacy is mainly left, concerning its protection, to legislation and service providers' self-regulation by means of privacy policies. However, all laws and codes of conduct are useless without enforcement. Based on this concept, this paper presents a framework conceived on the basis of privacy legislation. It uses a semantic model for the specification of privacy-aware data access rules and a middleware system which mediates between the service providers and the data sources and caters for the enforcement of the regulatory provisions.**

## I. Introduction

ON *the Internet, nobody knows you are a dog*, according to the famous 1993 Pat Steiner cartoon in The New Yorker, which has been very frequently cited in order to emphasize the potential for anonymity and privacy that the Internet was supposed to offer. However, the reality seems to be rather different; in fact, more than a century after the first essay identifying that privacy as a fundamental human right was endangered by technological advances [1], never before in history the citizens have been more concerned about their personal privacy and the threats by the emerging technologies [2].

The potential impact of contemporary Information and Communication Technologies on the privacy rights of the users is regarded as being among their most evident negative effects. The advances in mobile communications, location estimation and sensing technologies, along with data storage and processing technologies, have expanded the sphere of electronic services' provision and digital facilities from the Web to pervasive smart environments. They create impressive perspectives of rich, highly personalized and coherent services, information and computation ubiquity and, thus, spur an information revolution that brings significant improvements of the citizens' quality of life. On the other hand, they pose serious risks on the privacy rights of the data subjects; the personal data collection scale is augmented, information access, processing, aggregation, combination and linking are facilitated and new, sometimes even more sensitive, types of data are collected. A stream of data about individuals pours into data warehouses, while personal information is increasingly viewed as a valuable financial asset which is a subject of trading.

The service providers usually express their practices by means of privacy policies. Privacy policies concern the formal specification of an organization's business practices regarding the collection and the consequent use of personal data. The privacy policies are supposed to be restricted according to fair information principles and to comply with the relevant legal framework. Privacy legislation dictates how personal data should be treated after their provision by the data subjects to service providers and other processing entities, defining in essence the requirements for the privacy-aware management of personal data through their whole life cycle.

The Platform for Privacy Preferences (P3P) W3C specification [3] has been the first initiative towards this direction, providing a way for a web site to encode its relevant practices and to communicate them to the users that visit the site. Since proposed, P3P has received broad attention from both industry and research community, but it has also been subject of criticism from the current technical work, e.g., [4]. The major issue with P3P is the lack of the mechanisms for the enforcement of the specified privacy policies. In essence, P3P formalizes privacy promises given to the users for fair information practices; nevertheless, after their disclosure to a service provider, there is no guarantee about the fate of a user's personal data. Besides, there are numerous cases where the real practices contradict to well-stated privacy policies, e.g. [5], [6].

The challenge of enforcing a privacy policy has been thoroughly examined and several different solutions have been proposed, e.g., by IBM [7], OASIS [8] and Hewlett Packard [9]. These frameworks mainly focus on enterprise environments and provide the means for the automation of the privacy policies enforcement. The means for achieving this is to apply privacy-aware access control mechanisms which enhance traditional Role-Based Access Control (RBAC) mod els with additional, privacy-related aspects, such as the pur

This work was partially supported by the EU IST project DISCREET.

pose for data collection, retention periods, users' consents, notifications, etc.

However, all these solutions have their weak points. First, although they manage to address the issue of privacy policies internal enforcement within an organization to a great extent, they fail in providing the necessary guarantees for fair information practices to the users. In fact, since an organization possesses some personal data, their use or abuse by means of processing and disclosure are still based on good intents. Misuse may occur by a malicious employee with legitimate access to sensitive data or by any form of direct access to the data that bypasses the privacy protecting system. Second, the privacy policies specified in the context of these frameworks cannot be efficiently audited and verified as far as their regulatory compliance and consistency is concerned. Even an organization with the best intentions may specify a privacy policy that is not legislation-proof. Third, the specification of complex privacy policies and the continuous process of keeping them up-to-date introduce significant economical, operational and administrative overhead to an organization.

In the light of the above issues, this paper proposes a framework for the enforcement of privacy policies by the providers of e-services that are based on the legislation. The main concept behind the framework is the formal and detailed codification and specification of the regulatory provisions by a Privacy Authority into a single privacy policy document and its automatic dissemination to the providers. This unique privacy policy constitutes the technical translation of the privacy principles and regulations and overrides any other privacy policy defined by a provider. For its enforcement, a middleware architecture is introduced, that acts as a three way privacy mediator between the law, the users and the service providers. Its main component is the D-Core Box, a privacy proxy installed at the service provider's premises but totally controlled by the Privacy Authority. The D-Core Box stores any personal data, keeping them separated from the provider. Access to the data is granted based on the legislation originated privacy policy, as well as the relevant preferences expressed by the users via an associated mechanism that the framework offers.

The remainder of this paper is structured as follows. Section II provides some insights on the legal aspects of privacy; it codifies the legal privacy principles that form the requirements for the proposed framework. Section III describes the Ontology of Privacy, the semantic information model that constitutes the basis of the proposed approach, since it contains the rules stemming from the legislation. In Section IV, the means for enabling the users to specify their privacy preferences are outlined. Section V describes the middleware architecture that undertakes the task of enforcing the privacy rules, both regulations- and user- originated. The paper concludes in Section VI.

## II. Personal Data Protection Legislation

The starting point to obtain a formal modeling of the privacy legislation and also to design a technology capable of being privacy compliant and at the same time ensuring enforcement of privacy law provisions is identifying the regulatory requirements to be complied with.

A significant milestone in the privacy literature has been the codification of the fundamental privacy principles by the Organization for Economic Co-operation and Development (OECD), in 1980 [10], as this codification lays out the basis for the protection of privacy. The OECD principles are reflected in the European Directive 95/46/EC [11], which enforces a high standard of data protection and it is the most influential piece of privacy legislation worldwide, affecting many countries outside Europe in enacting similar laws. It is particularized and complemented with reference to the electronic communication sector by the Directives 2002/58/EC [12] and 2006/24/EC [13], which impose explicit obligations on network and service providers to protect the privacy of users' communications. The European Directives constitute the basis for the following summary of the main regulatory data protection requirements to be taken into account for the specification of the proposed framework.

### *A. Regulatory Requirements*

The following requirements are stemming from the European personal data protection legislation:

*Lawfulness of the data processing* : The system should be able to examine whether the data processing complies with applicable laws and regulations.

*Purposes for which data are processed* : The system should provide the means for identifying the data processing purposes, which must be lawful and made explicit to the data subject (namely the subject whose data are processed). Moreover, it should be able to check these purposes to avoid that data processed for a purpose may be further processed for purposes that are incompatible with these for which data have been collected.

*Necessity, adequacy and proportionality of the data processed* : The system should be able to guarantee that only the data functional, necessary, relevant, proportionate and not excessive with regard to the sought processing purpose are processed.

*Quality of the data processed* : The system should provide that the data processed are correct, exact and updated. Inaccurate data must be deleted or rectified; outdated data must be deleted or updated.

*Identifiable data* : The system should provide the means for keeping the data processed in identifiable form only for the time necessary to achieve the sought processing purpose.

*Information to the data subjects; consent and rights of the data subject* : The system should be able to provide for informing the data subject that his data are processed according to applicable data protection legislation. Moreover, the system should guarantee that when requested by applicable data protection legislation, the data subject's consent to the data processing is required, and that the data processing is performed according to the preferences expressed by the data subject. In addition, the system should enable the data subject to exercise the rights acknowledged by applicable data protection legislation in relation to intervention in the data processing (for example the right to access data, to ask for data rectification, erasure, blocking, the right to object the data processing, etc.).

*Data security and confidentiality* : The system should be secure in order to guarantee the confidentiality, integrity, and availability of the data processed. Moreover, the system should provide that the listening, tapping, storage or other kinds of interception or surveillance of communications and the related traffic data may be performed only with the data subject's consent or when allowed by applicable legislation for public interest purposes.

*Traffic data and location data other than traffic data; special categories of data* : The system should be able to guarantee that the processing of special categories of data (for example traffic or other location data, sensitive and judicial data) is performed in compliance with the specific requirements that the applicable data protection legislation sets forth for said categories of data.

*Access limitation* : The system should provide for an authorization procedure that entails differentiated levels of access to the data and for recording the accesses to the data.

*Data storage* : The system should be able to automatically delete (or make anonymous) the data when the pursued processing purpose is reached or in case of elapse of the data retention periods specified under applicable legislation.

*Notification and other authorizations from competent Data Protection Authority* : The system should be able to monitor compliance with the notification requirement and with the provisions on the authorizations of competent Data Protection Authority. Moreover, the system should provide for means that allow communications between the system and the competent Data Protection Authority.

*Supervision and sanctions* : The competent Data Protection Authority should be provided with the means for supervising and controlling all actions of personal data collection and processing.

*Lawful interception* : The competent public authority should be provided with the means to perform interception only when this is allowed by applicable laws and regulations and according to the conditions therein set forth. The necessary "hooks" for the lawful interception should under no circumstance become available to other not authorized third parties.

### B. The Regulations in a Nutshell

From the regulatory requirements presented above, the following facts are extracted:

*The role of the users* : The users are granted certain rights; the right to be informed regarding the collection or processing of personal data, to be asked about their explicit consent, to access their data. Additionally, they should be able to specify their privacy preferences and affect this way the service provision procedure, with respect to privacy.

*The role of the Authorities* : The legislation grants the Data Protection Authorities with certain rights and competences. These include the notification of the Authority, the supervision of the procedures and the means for performing Lawful Interception. That is, the Authority should be able to interact with the system.

*The role of semantics*: The semantics play a very crucial role in what can be characterized as "privacy context". On the one hand, each data item should be treated according to its special type. On the other hand, very important is the purpose for which the data are collected and processed.

*Access control*: The access to the data should be controlled. Beyond the legacy Role-Based Access Control models, decisions concerning access to personal data should take into consideration the semantics characterizing each privacy session.

*Complementary actions*: Access to the data should be accompanied by certain behavioral norms of the system. These include the information of the users or the request for their explicit consent, the notification of the Authorities, the automatic enforcement of data retention periods, as well as the adjustment of the detail level of the data.

*Security*: Naturally, in order for the personal data to be protected, the means for securing their transmission and storage should be taken; security always constitutes the bottom line for privacy protection and this paper takes as granted the availability of the corresponding means.

## III. Semantic Information Model

The modeling of the privacy legislation is achieved using a semantic information model that associates personal data, services and actors with explicitly defined regulatory rules. In that respect, the approach taken is to express any related information by means of an ontology, namely the Ontology of Privacy, which is implemented using the W3C Web Ontology Language (OWL) [14]. The vision is that the ontology should be as detailed as possible in terms of the various types of personal data and the types of services, so that the widest range of services and situations when personal data are involved can be covered. This is similar to what the Common Procurement Vocabulary (CPV) [15] represents for public procurement in Europe; it provides an exhaustive –almost semantic– list of several thousands of products that can constitute subject of public procurement.

In order to associate the personal data with specific processing tasks, the identification of the particular type of each personal data item is necessary. Moreover, in order to define the appropriate rules that will regulate the processing of a personal data item with respect to the purpose for which the information is provided by the user or requested by the service provider, a similar taxonomy of the provided services must be present. These taxonomies constitute separate subgraphs of the ontology. Therefore, the Ontology of Privacy provides a detailed vocabulary of personal data types and services' types, structured in an hierarchical way with well defined inheritance rules, that enables the system to associate all privacy related decisions to semantically specified notions. An equivalent taxonomy is needed for the involved actors; however, here we consider a very simple model, comprised of three actors: the user, the service provider and the Privacy Authority.

Regarding the personal data subgraph, all the types are defined as instances of the `PersonalData` OWL class. Inheritance hierarchies, as well as other relationships between personal data are defined using OWL properties. The first hierarchy specifies the inheritance of characteristics, referring to legislation-originating rules that regulate the collection and processing of personal data. The "root" personal data

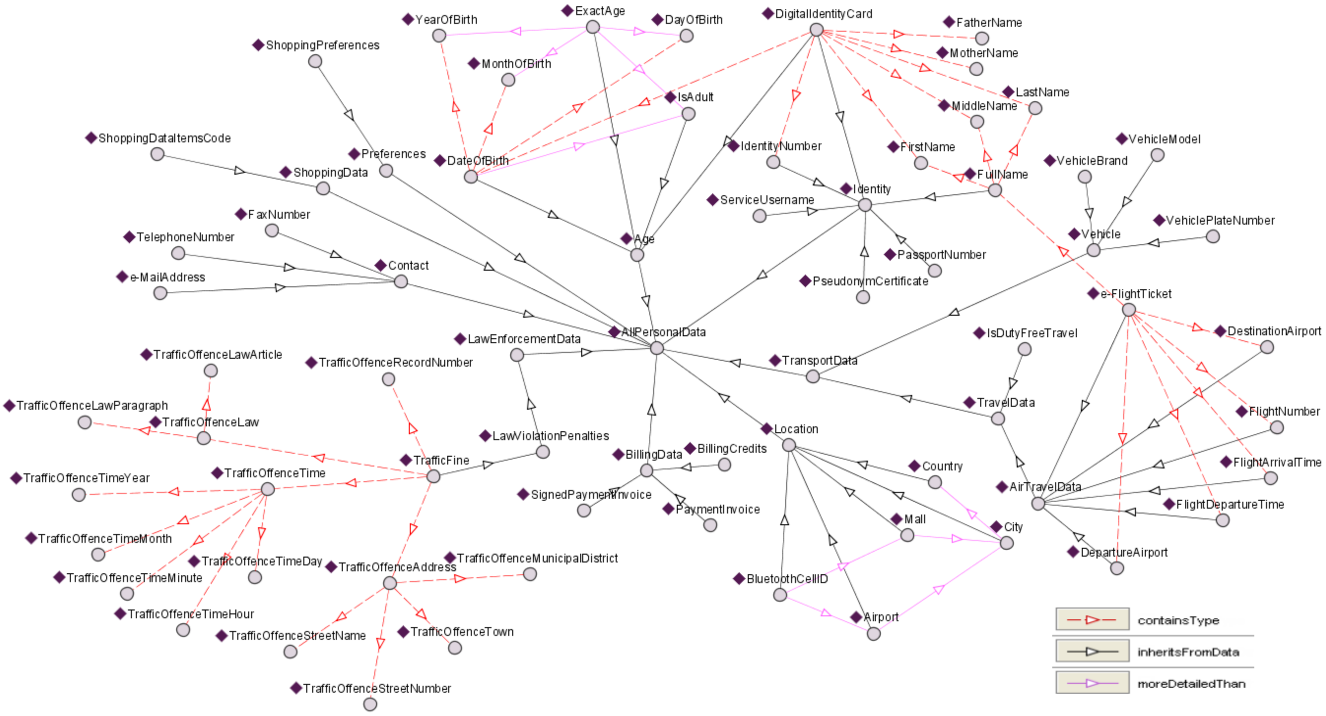


Fig 1: Ontology of Privacy – Personal Data Subgraph

type is the `AllPersonalData` type, from which all the other data types inherit, while "first level" children of `AllPersonalData` type instance include `Age`, `BillingData`, `Contact`, `Identity`, etc. These types constitute general data types, in essence categories of personal data types. This hierarchy is implemented by means of the `inheritsFromData` object OWL property.

The second hierarchy defined inside the `PersonalData` class deals with the detail level of personal data types. For this purpose, two properties are defined, `lessDetailedThan` and `moreDetailedThan`, being the one inverse to the other. In that respect, the `ExactAge` personal data types is `moreDetailedThan` the `YearOfBirth`, while the `Country` is `lessDetailedThan` the `BluetoothCellID`, with respect to the data subject's location.

The last relationship between the instances of the `PersonalData` class is the one that defines complex types resulting from simpler ones. In that respect, the data subject's

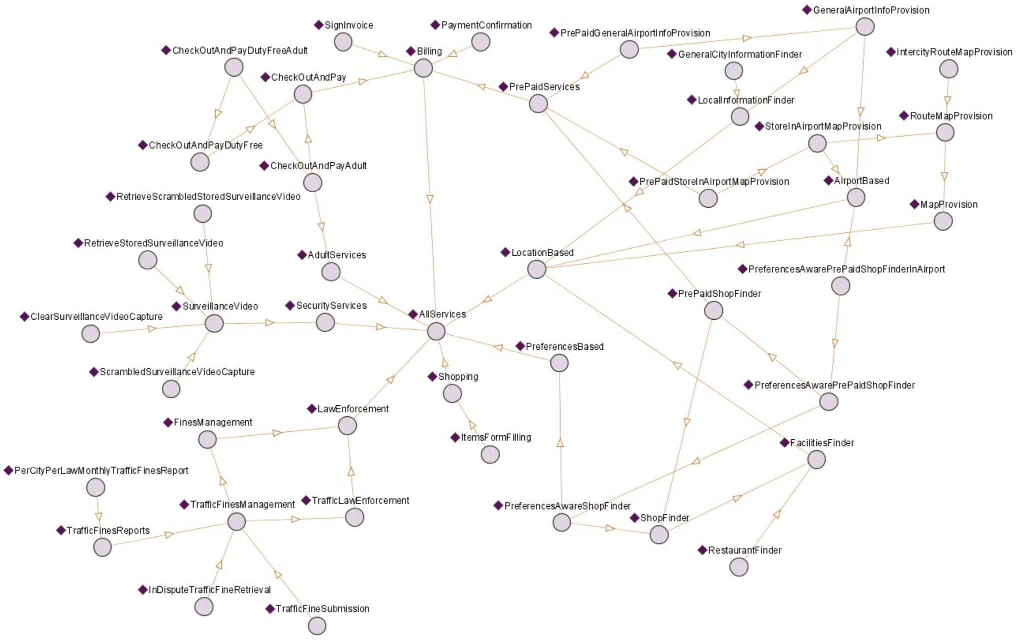

GeneralAirportInfoProvision
SignInvoice
PaymentConfirmation
PrePaidGeneralAirportInfoProvision
CheckOutAndPayDutyFreeAdult
Billing
GeneralCityInformationFinder
IntercityRouteMapProvision
CheckOutAndPay
PrePaidServices
LocalInformationFinder
StoreInAirportMapProvision
RouteMapProvision
CheckOutAndPayDutyFree
CheckOutAndPayAdult
PrePaidStoreInAirportMapProvision
AirportBased
MapProvision
RetrieveScrambledStoredSurveillanceVideo
RetrieveStoredSurveillanceVideo
AdultServices
LocationBased
PreferencesAwarePrePaidShopFinderInAirport
PrePaidShopFinder
SurveillanceVideo
SecurityServices
ClearSurveillanceVideoCapture
AllServices
PreferencesBased
ScrambledSurveillanceVideoCapture
Shopping
PreferencesAwarePrePaidShopFinder
LawEnforcement
FinesManagement
ItemsFormFilling
FacilitiesFinder
PerCityPerLawMonthlyTrafficFinesReport
TrafficFinesManagement
TrafficLawEnforcement
PreferencesAwareShopFinder
ShopFinder
TrafficFinesReports
RestaurantFinder
InDisputeTrafficFineRetrieval
TrafficFineSubmission

Fig 2: Ontology of Privacy – Subgraph of Services

FullName contains the FirstName, LastName and MiddleName data types. The containsType property and its inverse isContainedToType implement the corresponding relationships.

Fig. 1 illustrates part of the personal data subgraph, along with the OWL properties that implement the three types of relationships between the personal data instances, as described above.

The different services' types are organized as a hierarchy that defines inheritance of characteristics. All the defined types constitute instances of the Services OWL class. The "root" service type is the AllServices type, from which all the other services' types inherit, while "first level" children of AllServices type instance include AdultServices, Billing, LawEnforcement, LocationBased, etc. These types constitute general services' types. This hierarchy is implemented by means of the inheritsFromService object OWL property and its inverse one. It is noted that multiple inheritance is possible. As an example, a service can be location-based, while targeting adults; in this case, the service should inherit from both the LocationBased and the AdultServices types. Fig. 2 illustrates part of the services' subgraph; the arks represent inheritance associations, with the source node inheriting from the destination node.

As afore-mentioned, regarding the actors involved in the service provision chain, a very simple model has been considered. So far, the actors that have been defined are the DataSubject, the PrivacyAuthority and the ServiceProvider. However, this assumption can be easily removed with the extension of the Actors class to constitute a very detailed hierarchy of roles and –therefore– render the model fully role-based.

Access control rules are defined as instances of the Rules class of the ontology, in order to regulate the provision of services. Every rule is associated with a {personal data type, service type, actor} triad, using the corresponding refersToData, refersToService and refersToActor OWL object properties, and defines one or more properties that specify the permitted/forbidden actions of the *actor* over the *personal data type*, in the context of the provision of the *service type* under consideration, possibly along with certain complementary actions that must be additionally performed by the system.

With the use of OWL Annotation Properties, every rule contains the following information:

- DisclosureOfData: it defines whether the data of the specified type should be disclosed or not to the specified actor in the context of the provision of the specified service.
- RetentionPeriod: it specifies the period for which the data of the type under consideration should be retained.
- ModificationPermission: it defines if the specified actor should be granted with write/modify rights on the data of the specified type.

While the information above define the "core" of the rule, additional properties specify the complementary actions that should be potentially executed:

- DataSubjectInformation: it refers to the right of the user to be informed when the rule is applied (i.e., when in the context of the specified service, the personal data of the specified type are disclosed to the specified actor, or their modification takes place).
- DataSubjectConsent: it enables the user to be asked about explicit consent, prior enforce the body of the rule.
- AuthorityNotification: it forces the notification of the Authority when the rule is applied.

Finally, a rule may be characterized by certain meta-properties that serve for resolving conflicts between contradictory rules:

- appliesToPersonalDataDescendants: this binary property specifies whether the rule is inherited to the descendants of the specified data type, with respect to the corresponding subgraph of the ontology and the inheritance relationships.
- appliesToServiceDescendants: similarly to the case above, this binary property specifies the inheritance of the rule to the service type descendants.
- appliesToActorDescendants: although redundant since the corresponding actors' subgraph has not been defined yet, it refers to the inheritance of the rule to the descendants of the actor's type.
- OverrideDataSubjectPreferences: in certain cases, the user may have specified privacy preferences that contradict with the rules of the ontology; this property serves for defining which rule dominates over the other.

In Fig. 3, an example of an access control rule is illustrated. What this rule states is that "when the service under consideration is an adult service (AdultServices), and when the service provider (ServiceProvider) requests access to the personal data of IsAdult type (a binary data type, reflecting whether the data subject is an adult or not), the data should be given to the provider, while the data should not be further retained. The rule applies for the descendants of the AdultServices service type, while it does not apply for the descendants of the IsAdult personal data type and of the ServiceProvider actor type."

## IV. Specification of Privacy Preferences

While regulations-originated policies as specified in the Ontology of Privacy may determine the access permission to data up to some extent, the users should be able to determine the fate of their personal data. In that respect, a technical problem to be approached is how to enable the user to that direction, i.e., to control the disclosure, storage and processing of personal information, when the information is traveling through the various system and service components. To face this issue, it is necessary to associate to the data additional information which is communicated and stored with the data and brings information aimed at enforcing the specific treatment desired for the considered data.

Therefore, prior to leaving the user's terminal, the user's personal data are encapsulated into a data structure with the descriptive name Privacy Lock. The purpose behind its use is twofold: to make certain metadata (i.e., the user's preferences) available along with the respective data and to ensure the safe transmission of the data.

In essence, the Privacy Lock constitutes a secure shell that encapsulates the personal data transmitted by the terminal to the D-Core Box and vice versa, along with their metadata into an encrypted and optionally digitally signed object that ensures the safe communication. Moreover, the Privacy Lock can be used for the transmission of metadata solely, that express user preferences as far as either already stored data or data that will be processed in the future are concerned.

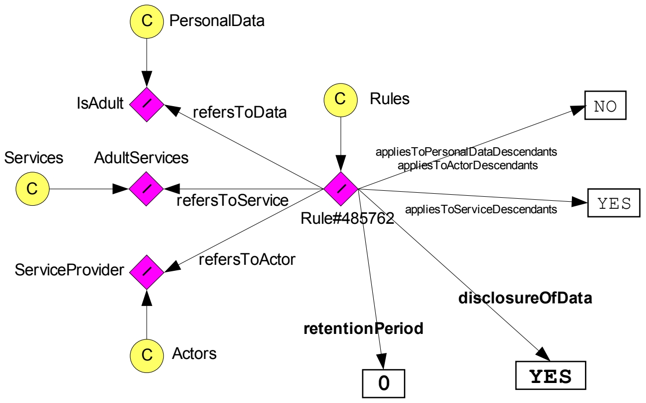


Fig 3: Ontology of Privacy – Example of Access Control Rule

An essential and mandatory attribute that is defined inside the Privacy Lock is the data type, which constitutes a critical parameter for their treatment in terms of disclosure, retention and processing. The type of the data is semantically specified with respect to the personal data subgraph of the Ontology of Privacy.

Apart from reflecting the data type, the metadata assigned to the data define certain properties for {personal data, services} [1] associations. These properties include:

- Whether the data should be disclosed for the provision of a certain service.
- The level of abstraction when the data are disclosed for the provision of a specified service.
- The expression of the user's preference to be informed or asked for consent whenever some processing or disclosure of personal data is about to take place.
- The determination of the desired retention period.
- Issues concerning data and metadata administration and management.

The metadata are formally expressed using a proprietary XML-based language, namely the Discreet Privacy Language (DPL). The DPL is used for the definition of all the necessary elements for the specification of all the afore-described types of metadata. Additionally, it is used for structuring the personal data when communicated from the user's terminal to a D-Core Box and vice-versa into a Privacy Lock, along with their privacy-related metadata.

The DPL's syntax is XML-based and contains the appropriate elements for the specification of the personal data attributes. In that respect, the DPL defines elements for the specification of:

- The personal data types, data items and service types that are regulated by the considered rule.
- Whether the considered rule is inherited by the personal data or service type descendants, with respect to the Ontology of Privacy.
- The rules themselves. The different rules' types defined include the disclosure level along with the corresponding level of precision, the access rights to the personal data, the demand for notifications and consents and the retention/validity period of the data.
- Meta-rules for the resolution of conflicts that naturally occur (e.g., contradictory user and regulations originated rules, rules overriding, etc.).

The detailed description of the DPL is beyond the scope of this paper; the normative definition of the DPL by means of Augmented Backus-Naur Form (ABNF) [16] specification is provided in [17].

## V. Middleware Architecture

The privacy protecting system takes the form of a distributed middleware that regulates the diffusion of personal data from the user towards the service provider, using legislative input. This "privacy broker" is comprised of three high level entities, each assigned to one of the three actors, i.e., the users, the service providers and the Privacy Authority. These entities form a privacy domain, the D-Core, inside which personal data handling is subject to both legislative requirements regarding privacy and user privacy preferences. The high level entities and the privacy domain they define are illustrated in Fig. 4 .

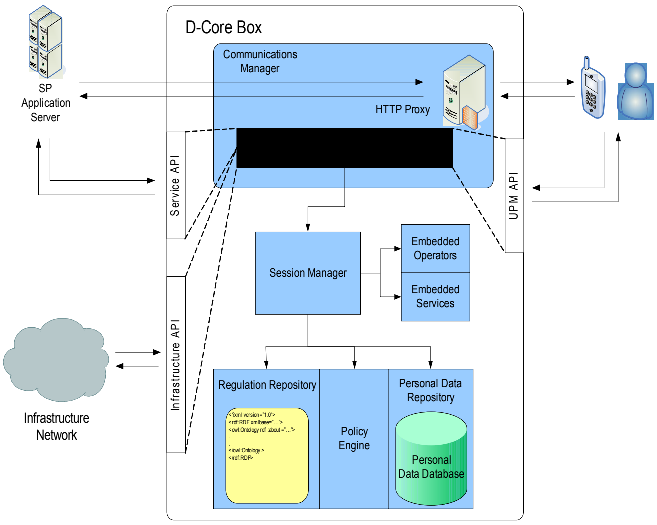
The entity that delivers the core system functionality is the D-Core Box ( Fig. 5 ). This is an intelligent privacy proxy, which, despite the fact that it is logically and physically deployed at the service provider's premises, is being managed by the Privacy Authority and not by the service provider. It constitutes the "edge" module of the D-Core and the border between the service provider's applications and the D-Core.

Any personal data provided by the user to the service provider are stored by the D-Core Box inside the Personal Data Repository. That is, the personal data are kept isolated from the service provider, which has no direct access to them. The storage may be short-time (e.g., immediate service provision) or long-time (e.g., services that require information archives). The data are stored together with the associated privacy preferences of the user, which are either transmitted with the data by means of a Privacy Lock, or defined/updated at any time through the corresponding interface that the D-Core Box offers. The same interface enables user to maintain complete control over the data, i.e., to update or delete them.

[1] It is noted that the metadata specification is actors-unaware; in fact, it is impossible for the user to refer to the internal structure of an organization.

When a service provider submits a request for users' personal data, the request is evaluated by the D-Core Box. All related decisions concerning personal data handling are taken by the Policy Engine which uses two sources of rules. The first source, the Ontology of Privacy is deployed to the D-Core Box by the Privacy Authority and is stored in the Regulations Repository. It provides the legislation-originated rules that are translated internally into a set of concrete DPL rules prior to be provided to the Policy Engine. The second source is the set of user defined privacy preferences which are provided by the Personal Data Repository, expressed as DPL rules. Through this prism, the Policy Engine examines the request and decides about the disclosure of the personal data and the potential execution of associated actions, such as the obfuscation of the data, the information of the user, the request for the user's consent, the notification of the Privacy Authority.

The key idea for the operation of the D-Core Box is to minimize the amount of personal data that are delivered to the application, without degrading the service. Moreover, the data should be disclosed pseudonymized or anonymized. In that respect, data that identify the user should not be disclosed, unless absolutely essential for the provision of the service. Therefore, in order to further minimize the amount of disclosed data, the D-Core Box incorporates modules that execute internally either simple data processing tasks or whole services' parts. These are, respectively, the Embedded Operators and Embedded Services components. Typical Embedded Operators functionalities are the filtering of the data precision prior to their disclosure (e.g., the translation of exact location to more abstract terms, or the transformation of the `ExactAge` data type to the `IsAdult` one.). Embedded Services undertake the execution of standard service components internally, mainly involving data that identify the user (e.g., e-mail sending or service charging mediation). This way, typical processing procedures that concern critical personal data, such as someone's identity or credit card number, are executed inside the D-Core Box and the need for the re spective personal data disclosure is eliminated. The Embedded Operators and Services are invoked by the Session Manager, a “stateful” component of the D-Core Box which orchestrates the functionality of the other components and manages every privacy session.



```
<?xml version="1.0">
<rdf:RDF xmlbase="...">
<owl:Ontology rdf :about ="...">
.
.
</owl:Ontology >
</rdf:RDF>
```

Fig 5: The D-Core Box

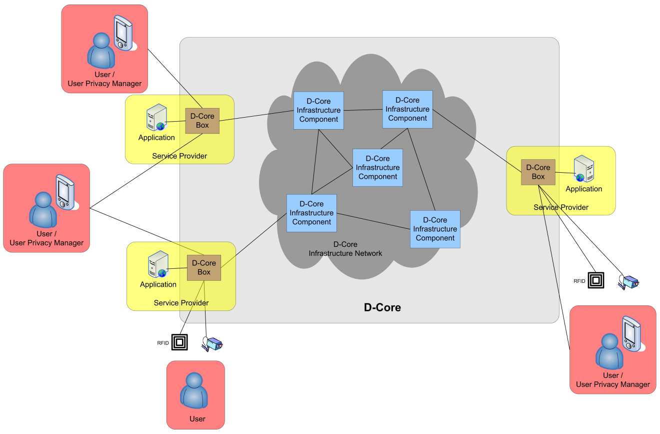


Fig 4: High level entities and the D-Core domain.

The D-Core extends at the user side with the User Privacy Manager (UPM). The UPM is a privacy agent for the user. It manages user identities for different services, it provides a console to edit privacy preferences and UIs to insert/edit personal data inside each identity and it creates Privacy Locks when personal data need to be delivered to the D-Core. The UPM is the peer entity of the D-Core Box for functions like informing the user when a service requests access to a specific personal data type, requesting user permission for this access and when creating and sending Privacy Locks containing personal data along with associated metadata.

The third entity in the D-Core domain is the Infrastructure Network. This is comprised of Infrastructure Components that constitute the Privacy Authority’s entry point to the domain. It provides to the Privacy Authority the means for the management of the Ontology of Privacy, the monitoring, and management of the system and the conduction of Lawful Interception. When the Ontology of Privacy is updated (e.g., due to legislation modification), then the updated version is transmitted through the Infrastructure Network to all the D-Core Boxes in order to consider the new requirements in the subsequent personal data requests. Regarding system management and monitoring, each Infrastructure Component undertakes the administrative responsibility regarding a number of D-Core Boxes and fulfills typical functions like collecting log data, checking status, generating error alarms, etc.

The communication of the D-Core Box with the other components of the D-Core domain as well as the applications is performed through a messaging framework, based on SOAP. The patterns defined for these interactions are presented in detail in [17], along with the detailed specification of the D-Core, its components and the respective interfaces between them and with the providers’ applications.

## VI. Conclusion

In this paper, a framework defining a protection domain for personal data was presented. Conceived on the basis of the legislation provisions, it provides the means for their enforcement and, therefore, the applications' commitment to adhere to privacy requirements. It presents two innovative features.

The first is the formal modeling of the data protection legislation in terms of the Ontology of Privacy. Using the Ontology of Privacy as a powerful tool for expressing the respective notions, a mere dictionary of terms is defined and shared by all system components and actors, starting from a Privacy Authority that specifies and configures the Ontology of Privacy on a constant basis and ending in the data subject and the service providers that make use of it. In that respect, all personal data are semantically marked and their type affects their consequent treatment by all the involved entities. Similarly, the specification of each service type provides the means for disclosure and processing purposes' specification and binding. The policies inside the ontology not only determine the necessity of a personal data type for a service's provision, but also constitute a complete set of regulations that are translated to access rights, services' flows and other rules for the protection of the personal data.

The second contribution of the proposed framework is the explicit separation of the personal data from the service providers' applications. With the mediation of D-Core Box, a service provider cannot gain access to personal data other than the one specified by the legislation and the user's preferences. The incorporation of several privacy-critical processing functionalities in the D-Core Box eliminates further the danger of data misuse.

## References

[1] S. D. Warren and L. D. Brandeis, "The Right to Privacy", *Harvard Law Review,* Vol. IV, No. 5, pp. 193–220, Dec. 1890.

[2] The European Opinion Research Group, "European Union citizens' views about privacy", *Special Eurobarometer 196,* Dec. 2003.

[3] The World Wide Web Consortium (W3C), The Platform for Privacy Preferences (P3P) Project, online: http://www.w3.org/P3P/.

[4] E. Bertino, J. Byun and N. Li, "Privacy-Preserving Database Systems", *Foundations of Security Analysis and Design III,* Lecture Notes in Computer Science 3655, Springer-Verlag, 2005.

[5] U.S.A. Federal Trade Commission, "Eli Lilly Settles FTC Charges Concerning Security Breach", FTC File No. 012 3214, Jan. 2002.

[6] U.S.A. Federal Trade Commission, "FTC Sues Failed Website, Toysmart.com, for Deceptively Offering for Sale Personal Information of Website Visitors", FTC File No. 002 3274, Jul. 2000.

[7] P. Ashley, S. Hada, G. Karjoth, C. Powers, M. Schunter, "The Enterprise Privacy Authorization Language (EPAL), *EPAL* 1.2 Specification", IBM Research Report, 2003.

[8] Organization for the Advancement of Structured Information Standards, "eXtensible Access Control Markup Language TC", 2004.

[9] M, Casassa Mont and R. Thyne, "A Systemic Approach to Automate Privacy Policy Enforcement in Enterprises, in *Proc. of 6th Workshop on Privacy Enhancing Technologies,* Lecture Notes in Computer Science, Vol. 4258, Springer-Verlag, 2006.

[10] Organization for Economic Co-operation and Development (OECD), "Guidelines on the Protection of Privacy and Transborder Flows of Personal Data", Sep. 1980.

[11] European Parliament and Council, "Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data", OJEC, No. L 281, pp. 31-50, Nov. 1995.

[12] European Parliament and Council, "Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector", OJEC, No. L 201, pp. 37-47, Jul. 2002.

[13] European Parliament and Council, "Directive 2006/24/EC on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC", OJEC, No. L 105, pp. 54-63, Apr. 2006.

[14] The World Wide Web Consortium (W3C), "Web Ontology Language (OWL)", online: http://www.w3.org/2004/OWL/.

[15] European Parliament and Council, "Regulation 2195/2002/EC on the Common Procurement Vocabulary (CPV)", Official Journal of the European Communities, No. L 340, pp. 1–562, December 2002.

[16] D. Crocker, P. Overel. "Augmented BNF for Syntax Specifications: ABNF," RFC2234, IETF, Nov. 1997.

[17] Georgios V. Lioudakis et. al., "Implementation Report on the Core System", IST DISCREET Deliverable D3102, January 2008.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Access Control Models in Heterogeneous Information Systems: from Conception to Exploitation

Aneta Poniszewska-Maranda
Institute of Computer Science,
Technical University of Lodz, Poland
anetap@ics.p.lodz.pl

***Abstract*—The development of the information systems should answer more and more to the problems of federated data sources and the problems with the heterogeneous distributed information systems. The assurance of data access security realized in the cooperative information systems with loose connection among local data sources is hard to achieve mainly for two reasons: the local data sources are heterogeneous (i.e. data, models, access security models, semantics, etc.) and the local autonomy of systems does not allow to create a global integrated security schema.**

**The paper proposes to use one common set of access control concepts to support the access control management in security of heterogeneous information systems. The UML (Unified Modelling Language) concepts can be used to define and implement the most popular access control models, such as DAC, MAC or RBAC. Next, the concepts derived from different models can be joined to use one common approach comprehensible for each administrator of each cooperative information system in the federation.**

## I. Introduction

THE development of the information systems should answer more and more the problems of federated data sources and the problems with the heterogeneous distributed information systems. It is necessary to solve the problems with structured or semantic conflicts on the level of information stored in the systems, to assure the acknowledgment of security constraints defined for the local information sources and to create the control process on the global level of cooperative information systems. We do not solve all these problems in this paper. We only propose to use one common set of access control concepts to support the access control management in the security of heterogeneous information systems. The UML (Unified Modelling Language) [1], [2] concepts can be used to define and implement the most popular access control models, such as DAC (Discretionary Access Control), MAC (Mandatory Access Control) or RBAC (Role-based Access Control) [3], [4], [5], [6]. Next, the concepts derived from different models can be joined to use in the security management one common approach comprehensible for each administrator of each cooperative information system in the federation.

The assurance of the data access security realized in the federated information systems with loose connection among local data sources is hard to achieve mainly for two reasons: the local data sources are heterogeneous (i.e. data, models, access security models, semantics, etc.) and the local autonomy of systems does not allow to create a global integrated security schema. Each of systems, subsystems or applications of the federated information system can be secured by a different security policy, and one common approach joining the concepts from different policies can help in the process of security policy integration on the global level.

The paper is structured as follows: the first part presents the access control in heterogeneous information systems and the creation process of security scheme. The second part deals with the connection of UML concepts with the concepts derived from three types of access control models, i.e. DAC, MAC and the extended RBAC. The third part describes the creation of user profiles based on presented access control models. Finally, the fourth part presents the common access control approach comprehensible for security administrators of each information system in the federation.

## II. Access control in heterogeneous information systems

The security policies of a system generally express the basic choices made by an institution for its own data security. They define the principles on which access is granted or denied. The access control imposes constraints on what a user can do directly, and what the programs executed on behalf of the user are allowed to do. A security access system can be defined by using two parts that cooperate with each other: the security access strategy, which describes all the environments and the specifications of the entire organization on the security level (i.e. organizational and technical aspects), and the access model with:

- a set of concepts to describe objects (data access) and subjects (users),
- a definition of the users' access rights to data,
- an access control policy which describes how users can manipulate data, defines data structure and administers the user' access rights to data.

Two categories of security policies of the information systems can be distinguished: discretionary security policy or mandatory (non-discretionary) security policy. It is possible to find access control models based on these policies:

- *Discretionary Access Control (DAC)* model [3] manages the users' access to the information based on the user' identification and on the rules defined for every user (subject) and object in the system using the access control matrix. For each subject and object in a system there are authorization rules that define the access modes of the subject on the object.
- *Mandatory Access Control (MAC)* model [3] is based on the classification of subjects and objects in the system. Each subject and each object is attached to a security level, which is composed of a classification level and a category. The classification levels are arranged by their sensibility degree: Top Secret, Secret, Confidential and Unclassified. In this model each subject has his own authorization level that allows him the access to the objects starting from the classification level, which has lower or equal range.
- *Role-Based Access Control model (RBAC)* model [4], [5], [6] requires the identification of roles in a system. The role is properly viewed as a semantic structure around which the access control policy is formulated. The role can represent the competency to do a specific task and it can embody the authority and responsibility of the system users. The permissions are associated with roles and the users are assigned to appropriate roles. The roles are created for various job functions in an organization and the users are assigned to roles based on their responsibilities and qualifications. The user playing a role is allowed to execute all access modes to which the role is authorized. The user can take different roles on different occasions.
- *Extended RBAC (eRBAC)* model [7]—each role realizes a specific task in the enterprise process and it contains many functions that the user can take. For each role it is possible to choose the necessary system functions. Thus, a role can be presented as a set of functions that this role can take and apply. Each function can have one or more permissions, and consequently a function can be defined as a set or a sequence of permissions. If an access to an object is required, then the necessary permissions can be assigned to the function to complete the desired job. Specific access rights are necessary to realize a role or a particular function of this role. These rights determine the possibility to execute an application or to access necessary data, and moreover they correspond with these functions. Thus, the specific permissions have to be defined for each function (Fig. 1).

The objectives of the security policy in cooperative information systems are to respect the local security model of each system (each model specifies the security principles of a local system) and to control the indirect security that comes from global cooperation level: a member of a local system may in another local system access only the equivalent information according to his local profile. It is possible to find the situations in which some information systems have to cooperate with each other creating the set of cooperative information systems. Each system can have another security policy for describing the access control rules to access its data. This situation can involve some difficulties and heterogeneities in definition of the global security model. The following types of global security heterogeneities were found [8]:

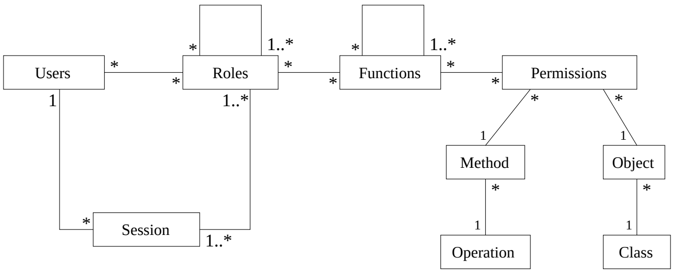


Fig. 1. Extension of the RBAC model

- heterogeneity of the information system security strategies (centralized vs. decentralized authorization, ownership vs. administration paradigm, etc.),
- heterogeneity of security policies between MAC models, DAC models, RBAC models and their extensions,
- different kinds of access rights (positive, negative or mixed), different authorization units (subjects, users, group, roles), different access administration concepts (Grant, Revoke, etc.),
- heterogeneity of security entities: elements of security concept model (databases, domain, types/classes/relations or object, etc.) between local schemes.

This paper deals with the second point—the problem of heterogeneity of security policies. Each system in the federation can use another security policy and in consequence another access control model. The possible solution is to find the common concepts in order to connect all security concepts from different models. It should be possible to present different elements from different access control models using one common approach—the ideas that are universal for all the models. This way, the access control in the heterogeneous information systems can be presented using one common set of concepts in order to manage such system in a common and simpler way.

Two types of actors cooperate in the creation stage of an information system and the security scheme associated with it [9]: the information system developer who knows specifications of an information system that need to be realized and the security administrator who has the knowledge of the general security policy that should be respected on the enterprise level.

The creation process of the security scheme in heterogeneous information systems is proposed as follows (Fig. 2):

- Application developer creates the system application or a set of system applications using the UML concepts. UML is used to define the application Model containing all elements that express the needs of the users.

- Application developer initiates the process of user profile creation (e.g. role engineering for eRBAC model) [10], [9] based on the security rules concerning this application.
- The application Model created by the developer is translated to the concepts of access control models (e.g. DAC, MAC or eRBAC) based on the connection of UML concepts with the concepts of DAC/MAC/eRBAC model (Section 3). Also the process of user profile creation is finished on the developer level (Section 4).
- Security administrator receives the Model containing the lists of access control elements which are presented in a special form, e.g. in the XML files. The administrator finishes the process of user profile creation using the rules of the global security policy (Section 4).
- The application Model together with its associated access control rules (defined by the developer and by the administrator) are transformed to the common access control Model using the common concepts for the heterogeneous security systems (Section 5). The results of this transformation are given in the XML files with its associated DTD files which are legible and comprehensible for each security administrator of each information system in the federation.
- Security administrator(s) can manage the federation of the heterogeneous information systems on the access control level using the common set of concepts.

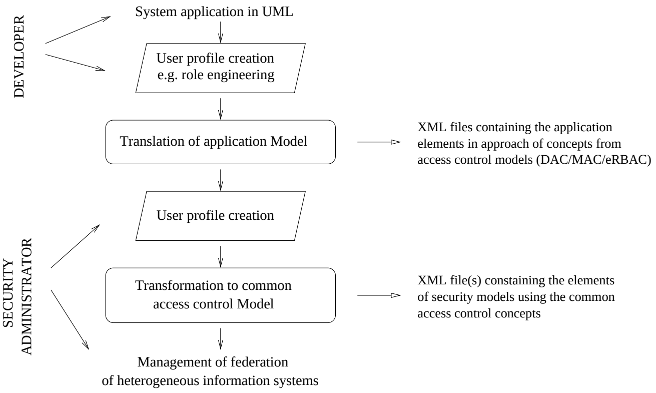

DEVELOPER
System application in UML
User profile creation e.g. role engineering
Translation of application Model
XML files containing the application elements in approach of concepts from access control models (DAC/MAC/eRBAC)
User profile creation
SECURITY ADMINISTRATOR
Transformation to common access control Model
XML file(s) constaining the elements of security models using the common access control concepts
Management of federation of heterogeneous information systems

Fig. 2. Creation process of security scheme in heterogeneous systems

The next sections presents the theoretical groundwork for the main stages of this process.

## III. Connection of UML concepts with concepts of access control models

The methods of the object-oriented analysis and conception can be considered an evolution of the systemic approach towards greater coherence among the information system objects and their dynamics. The Unified Modelling Language can be mentioned in this category as a standard for the object-oriented modelling in the field of software engineering [1], [2]. It contains a suite of diagrams for requirements, analysis design and implementation phases of the development of systems. Due to the diagrams, it is possible to visualize and manipulate the modelling elements. Out of different types of diagrams defined by the UML and representing different viewpoints of the modelling, three types, i.e. class diagram, use case diagram and interaction diagram, are in the focus of attention of the presented study. The UML has been chosen for the representation of security models because nowadays it is a standard tool, properly reflecting the description of the information system and its needs. UML gives the possibility to present the system using different models. The purpose is to define and implement the access control models, such as DAC, MAC and extended RBAC with the use of UML. To achieve this, the UML concepts and concepts from three models should be joined.

### A. *Concepts of DAC model associated with UML concepts*

The elements of the UML class diagram and interaction diagram, e.g. sequence diagram, can be used to define the concepts of the DAC model (Fig. 3):

- *DAC subject* (user) is found in the instance of actor class from UML class diagram supported eventually by UML constraints,
- *DAC object* defined by UML object,
- *DAC operation* defined by UML operation and furthermore
- user identification can be described by UML constraints.

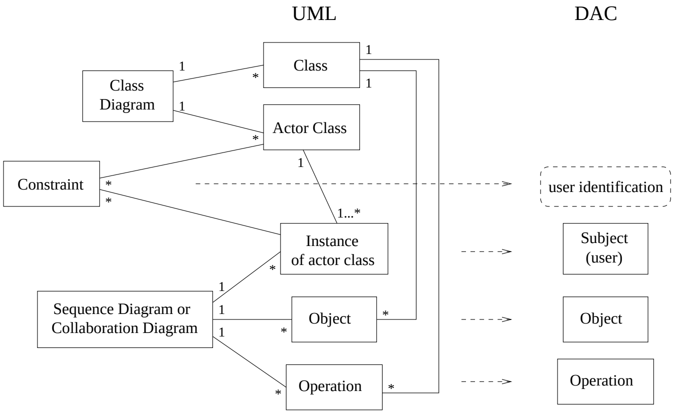


Fig. 3. UML concepts and their relationships with DAC model

### B. *Concepts of MAC model associated with UML concepts*

Similarly, the elements of the UML class diagram and interaction diagram, e.g. sequence diagram, can be used to find the elements of the MAC model (Fig. 4):

- *MAC subject* is defined by the instance of actor class from UML class diagram supported eventually by UML constraints,
- *MAC object* defined by UML object,
- *MAC operation* defined by UML operation and furthermore
- authorization level can be represented by UML stereotypes or element properties,

- security level, i.e. classification level and category, presented by UML element properties or by UML constraints.

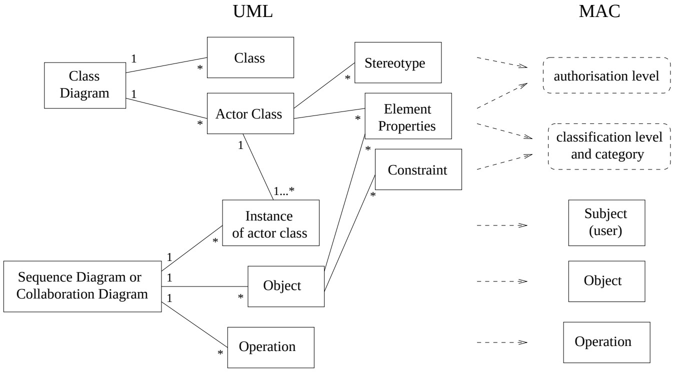


Fig. 4. UML concepts and their relationships with MAC model

### C. Concepts of extended RBAC model associated with UML concepts

Two types of the UML diagrams have been chosen to present and implement the elements of the extended RBAC model: use case diagram and sequence diagram (Fig. 5). The presentation of connections between the UML and the extended RBAC concepts is described widely in [7]:

- *RBAC role* is joined with UML actor,
- *RBAC function* joined with UML use case,
- *RBAC methods* and objects with methods and objects of UML,
- *RBAC permissions* can be found in the interaction diagrams,
- *RBAC constraints* are joined with constraint concept existing in UML [10],
- relations of different types that occur between the elements of the extended RBAC model can be found in the use case diagrams and in the interaction diagrams.

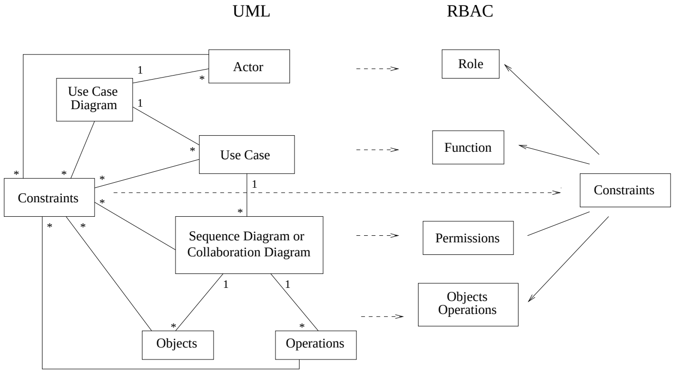


Fig. 5. UML concepts and their relationships with extended RBAC model

## IV. Creation of user profiles based on access control models

System security policies demand that for each user a set of operations that he will be allowed to execute should be clearly defined. Consequently, for each user a set of permissions should be defined. It suffices to specify the permissions for the execution of certain methods on each object accessible for that user. According to the connections between UML and three access control models given above, a definition of user profiles on the developer and administrator levels can be proposed.

### A. DAC model

The authorization rules for the subjects and objects in the DAC model using the UML concepts are defined with the use of class diagram, from which it is possible to obtain the list of system subjects, i.e. system users. The users have the authorizations to realize different operations (i.e. read, write, execute, own) on different objects depending on the definitions of elements situated in the class diagram or in the interaction diagrams in which these subjects participate. These types of UML diagrams allow obtaining the list of objects and the list of operations realized on these objects for each subject in an information system. The user identification is realized by the UML constraints attached to the classes in a class diagram, or to the actors or objects in the interaction diagrams. The notion of ownership policy, characteristic for the DAC model, can be determined with the use of direct or indirect relations between objects in the interaction diagrams or between their classes in the class diagram.

### B. MAC model

The implementation of the MAC model using the UML concepts is realized also with the use of the class diagram and the interaction diagrams, which allow to obtain the list of the system users (i.e. system subjects) and the lists of the system objects on which these users can perform different operations. The characteristic notions for the MAC model, such as authorization level and security level (i.e. classification level and category) can be obtained using the UML extension mechanisms, such as stereotypes, element properties or constraints, defined for the model elements in the class diagram or in the interaction diagram.

Both in the DAC model as well as in the MAC model, the definition of the user's authorizations to execute some operations on different objects (i.e. the definition of the user's permissions) with all the model characteristic notions can be realized by the system developer. The security administrator is responsible for any changes in these definitions that can be made by the new assignments of the information system elements or addition of the new constraints defined for the system elements.

### C. Extended RBAC model

The implementation of the extended RBAC model using the UML concepts is realized with the use of the sequence diagrams, where permissions are assigned to the rights of execution of the methods realized in each use case [7]. The UML meta-model is applied to define the roles of the RBAC model, the functions that are used by these roles to co-operate with the information system and the permissions needed to realize these functions. Owing to the use case diagrams a list of actors co-operating with the information system is obtained. An analysis of these diagrams allows the automatic specification of relations of the following types: R-R (role-role) relation (with the use of the generalization relation between the actors), R-F (role-function) relation (with the use of the association relation between the actors and the use cases) and F-F (function-function) relation (the generalization relation between the use cases). The description of a use case using the interaction diagrams (e.g. sequence diagrams) presents activities needed to realize the functions of a system. Each activity is a definition of execution of a method on an object. Therefore the F-P relations can also be automatically managed. Our definition of a set of roles in an information system with the use of the UML diagrams contains two stages: assignment of a set of privileges (permissions) to the use case in order to define the function and assignment of a set of use cases (functions) to the actor in order to define the role.

In order to create a set of roles assigned to a user profile, users should be assigned to roles. This stage is realized by the security administrator.

## V. Common concepts for heterogeneous information systems

In an information system the access control is responsible for granting direct access to system objects in accordance with the modes and principles defined by protection policies. An access control system defines: the *subjects* (active entities of a system) that access the *information* (passive entities) executing different *actions*, which respect the access rules. The subjects can describe the *users* or the processes that have access to the data stored in a system. The information, i.e. the data, determines the system *objects* on which the actions represented by the most popular *operations*, i.e. read, write, delete, execute, can be performed. Therefore, it is possible to distinguish three main sets of elements describing the access control rules: **subjects**, **objects** and **operations**.

We propose to represent the elements of access control models, i.e. DAC, MAC and eRBAC, using these three sets and additionally the concept of constraints (Fig. 6). A security constraint is an information assigned to the system elements that specifies the conditions to be satisfied so that the security rules and global coherence of a system can be guaranteed [10].

This connection is possible with regards to the features of access control concepts [3] and the concepts of access control models (Section 2). It can be realized by the automatic transformation of XML files, containing the application elements in approach of concepts of access control models (DAC/MAC/eRBAC), to the XML file(s) containing these elements using the common concepts. It is necessary to create the DTD (Data Type Definition) files for these XML files to define their structures. The root elements of DTD files for each access control model are given as follows:

- DTD file for DAC model:
$< !ELEMENT\ DAC(user+, object+, operation+, userIdentification*) >$

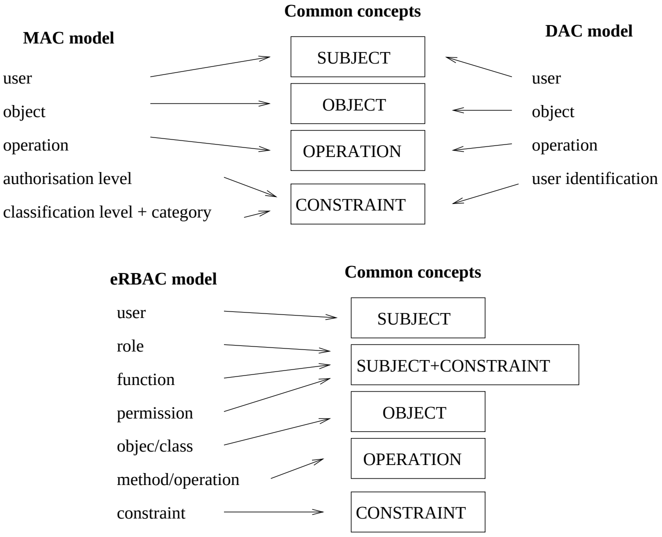


Fig. 6. Common concepts for access control models

- DTD file for MAC model:
$< !ELEMENT\ MAC(user+, object+, operation+, authorizationLevel*, classificationLevel*, category*) >$
- DTD file for eRBAC model:
$< !ELEMENT\ eRBAC(user+, role+, function+, permission+, method+, object+, operation+, class+, constraint*) >$

The root element of DTD file containing the common concepts for describing the security elements of each access control model is as follows:

$< !ELEMENT\ commonModel(subject+, object+, operation+, constraint*) >$

It describes the common concepts of heterogeneous security systems. The XML files based on such DTD file are intended for security administrator(s) to manage the federation of heterogeneous security systems.

## VI. Conclusions

The paper describes the creation and management of security scheme in heterogeneous information systems on the access control level. We propose to use one common set of access control concepts to support the access control management in heterogeneous security systems.

The UML concepts are proposed to support the access control management in the information system security. UML can be used to realize the access control models as DAC, MAC or the extension of the classic RBAC model and next to help in the creation of user profiles based on these access control models. UML, a standard language for object analysis and design nowadays, has been chosen in view of the fact that it enables the complex presentation of the information system and different aspects of information system security. The implementation of access control models is realized using the UML concepts connected earlier with the concepts of these models.

The complexity of the information systems generates the need for new more effective techniques and tools. The object approach and modelling using UML gives the possibility to create the security scheme for the heterogeneous information systems that accepts different access control models, such as DAC, MAC or extended RBAC model.

## References

[1] G. Booch, J. Rumbaugh, and I. Jacobson, "The unified modeling language user guide," *Addison Wesley*, 1998.

[2] O. M. Group, "Omg unified modeling language specification," *Reference Manual*, 2005.

[3] S. Castaro, M. Fugini, G. Martella, and P. Samarati, "Database security," *Addison-Wesley*, 1994.

[4] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[5] R. S. Sandhu and P. Samarati, "Access control: Principles and practice," *IEEE Communication*, vol. 32, no. 9, pp. 40–48, 1994.

[6] D. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed nist role-based access control," *ACM Transactions on Information and Systems Security*, 2001.

[7] A. Poniszewska-Maranda, G. Goncalves, and F. Hemery, "Representation of extended rbac model using uml language," *Proc. of SOFSEM 2005, LNCS 3381, Springer-Verlag*, 2005.

[8] E. Disson, D. Boulanger, and G. Dubois, "A role-based model for access control in database federations," *Information and Communications Security, Proc. of 3th ICICS, China*, 2001.

[9] A. Poniszewska-Maranda, "Access control coherence of information systems based on security constraints," *Proc. of 25th International Conference on Computer Safety, Security and Reliability, LNCS, Springer-Verlag*, 2006.

[10] ——, "Security constraints in access control of information system using uml language," *Proc. of 15th IEEE WETICE, England*, 2006.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Semantic Aware Access Control Model with Real Time Constraints on History of Accesses

Ali Noorollahi Ravari
Network Security Center, Computer Engineering Department,
Sharif University Of Technology, Tehran, Iran
Email: noorollahi@ce.sharif.edu

Morteza Amini, Rasool Jalili
Network Security Center, Computer Engineering Department,
Sharif University Of Technology, Tehran, Iran
Email: { m_amini@ce., jalili@} sharif.edu

***Abstract*—With the advent of semantic technology, access control cannot be done in a safe way unless the access decision takes into account the semantic relationships among the entities in a semantic-aware environment. SBAC model considers this issue in its decision making process. However, time plays a crucial role in new computing environments which is not supported in the model. In this paper we introduce the Temporal Semantic Based Access Control model (TSBAC), as an extension of SBAC, which enhances the specification of user-defined authorization rules by constraining time interval and temporal expression over users' history of accesses. A formal semantics for temporal authorizations is provided and conflicting situations (due to the semantic relations of the SBAC model and a sub-interval relation between authorizations) are investigated and resolved in our proposed model. An architecture for the access control system based on the proposed model is presented, and finally, we discuss and evaluate TSBAC.**

## I. Introduction

ACCESS control is a mechanism that allows owners of resources to define, manage and enforce access conditions applicable to each resource [1]. An important requirement, common to many applications, is related to the temporal dimension of access permissions. In these systems, permissions are granted based on previous authorizations given to the users of the system in specific time points (in the past).

Another critical requirement is the possibility of expressing the semantic relationships that usually exist among different authorization elements, i.e. subjects, objects, and actions. To overcome this challenge, our model is constructed based on the SBAC model [2, 3] which is a semantic-based access control model. SBAC authorizes users based on the credentials they offer when requesting an access right. Ontologies are used for modeling entities along with their semantic interrelations in three domains of access control, namely subjects domain, objects domain, and actions domain. To facilitate the propagation of policies in these three domains, different semantic interrelations can be reduced to the subsumption relation.

In this paper we unify the two concepts mentioned previously, that is, we use SBAC (as the base model), and associate a temporal expression with each authorization. Due to the nature of some application domains (such as the banking environment), a real representation of time is required to be used for modeling temporal dependencies between history of accesses. So, in this paper, we use real time operators to impose constraints on elements of History Base. Furthermore, a temporal interval bounds the scope of the temporal expressions (e.g., [1,20] shows that the authorization is valid for time interval starting at ‘1’ and ending at ‘20’). Thus, the main feature provided by TSBAC is the possibility of specifying authorization rules which express temporal dependencies among authorizations. These rules allow derivation of new authorizations based on the presence or absence of other authorizations in specific past time instants (stored in *History Base* in the form of done(t,s,o,a) and denied(t,s,o,a)

A formal semantics is defined for temporal authorizations. The subject of Temporal Authorization Base (TAB) administration and conflicting situations are investigated and resolved. An architecture for the access control system based on TSBAC, and an evaluation is presented.

The rest of this paper is organized as follows: in Section 2 we discuss the related works on this topic. Section 3 gives a brief introduction of the SBAC model and describes the model of time used throughout our work. In section 4, we represent our authorization rules in detail and offer the formal semantics followed by a brief description of administration of the authorization base and conflict resolution in access decision point. Section 5 gives an architecture for the access control system based on the proposed model. In section 6 we give a brief evaluation of our work. Finally, section 7 concludes the paper.

## II. Related Work

Access control systems for protecting Web resources along with credential based approaches for authenticating users have been studied in recent years [1]. With the advent of Semantic Web, new security challenges were imposed to security systems. Bonatti *et al.*, in[4] have discussed open issues in the area of policy for Semantic Web community such as important requirements for access control policies. Developing security annotations to describe security requirements and capabilities of web service providers and requesting agents have been addressed in [5]. A concept level access

978-83-60810-14-9/08/$25.00 © 2008 IEEE

control model which considers some semantic relationships in the level of concepts in the object domain is proposed in [6]. The main work on SBAC, which is the basis of our model, is proposed in [2, 3] by Javanmardi *et al.*. SBAC is based on the OWL ontology language and considers the semantic relationships in the domains of subjects, objects, and actions to make decision about an access request.

The first security policy based on past history of events was introduced as Chinese Wall Security Policy (CWSP) [7]. The objective of CWSP is to prevent information flows which cause conflict of interest for individual consultants. Execution history also plays a role in Schneider's security automata [8] and in the Deeds system of Edjlali [9]. However, such works focus on collecting a selective history of sensitive access requests and use this information to constrain further access requests; for instance, network access may be explicitly forbidden after reading certain files. Another approach which considers the history of control transfers, rather than a history of sensitive requests, is presented in [10].

In a basic authorization model, an authorization is modeled by a triple $(s,o,\pm a)$, interpreted as "subject *s* is (not) authorized to exercise access right *a* on object *o*". Recently, several extensions to this basic authorization model have been suggested. One of them is the temporal extension, which increases the expressive power of the basic authorization model [11-15]. In the model proposed by Bertino *et al.* in [11], an authorization is specified as $(time,auth)$, where $time=(t_b,t_e)$ as the time interval, and $auth=(s,o,m,pn,g)$ as an authorization. Here, $t_b$ and $t_e$ represent the start and end times respectively, during which *auth* is valid; *s* represents the subject, *o* the object, and *m* the privilege; *pn* is a binary parameter indicating whether an authorization is negative or positive, and *g* represents the grantor of the authorization. This model also allows operations *WHENEVER*, *ASLONGAS*, *WHENEVERNOT*, and *UNLESS* on authorizations. For example, *WHENEVER* can be used to express that a subject $s_i$ can gain privilege on object *o* whenever another subject $s_j$ has the same privilege on *o*. Later Bertino *et al.* in [14] extended the temporal authorization model to support periodic authorizations. They completed their research in [16] by presenting a powerful authorization mechanism that provides support for: (1) periodic authorizations (both positive and negative), that is, authorizations that hold only in specific periods of time; (2) user-defined deductive temporal rules, by which new authorizations can be derived from those explicitly specified; (3) a hierarchical organization of subjects and objects, supporting a more adequate representation of their semantics. From the authorizations explicitly specified, additional authorizations are automatically derived by the system based on the defined hierarchies.

## III. Preliminaries

In this section we give a brief introduction of the SBAC model, proposed by Javanmardi *et al.* [2, 3], and introduce the representation of time used throughout this work.

### A. Introduction to SBAC

Fundamentally, SBAC consists of three basic components: Ontology Base, Authorization Base, and Operations. Ontology Base is a set of ontologies: Subjects–Ontology (SO), Objects–Ontology (OO), and Actions–Ontology (AO).

By modeling the access control domains using ontologies, SBAC aims at considering semantic relationships in different levels of ontology to perform inferences to make decision about an access request. Authorization Base is a set of authorization rules in the form of $(s,o,\pm a)$ in which *s* is an entity in SO, *o* is an entity defined in OO, and *a* is an action defined in AO. In the other words, a rule determines whether a subject which presents a credential *s* can have the access right *a* on object *o* or not.

The main feature of the model is reduction of semantic relationships in ontologies to subsumption relation. Given two concepts *C* and *D* and a knowledge base $\Sigma$, $C<D$ denotes that *D* subsumes *C* in $\Sigma$. This reasoning based on subsumption proves that *D* (the subsumer) is more general than *C* (the subsumee).

By reducing all semantic relationships to the subsumption, the following propagation rules are enough:

- *Propagation in subjects domain*: Given $(si,o,\pm a)$, if $sj<si$ then $(sj,o,\pm a)$.
- *Propagation in objects domain*: Given $(s,oi,\pm a)$, if $oj<oi$ then $(s,oj,\pm a)$.
- *Propagation in actions domain*:
  - Given $(s,o,+ai)$, if $aj<ai$ then $(s,o,+aj)$.
  - Given $(s,o,-aj)$, if $aj<ai$ then $(s,o,-ai)$.

### B. Modeling of Time

In this paper, we assume a real representation of time. It is worthwhile to note that, we suppose that the response time of the access control system is trivial and thus we ignore the time duration required by the system to check whether a requested access is granted or denied. This assumption allows us to take an access request time as the access time recorded in the history.

A good representation of time for instantaneous events, if possible, is using an absolute dating system. This involves time stamping each event with an absolute real-time. For instance, a convenient dating scheme could be a tuple consisting of the *year*, *month in the year*, *day in month*, *hour in the day*, *minutes*, and *seconds*. For example, (2008 1 20 10 4 50) would be the 20th day of January 2008, at 10:04 (AM) and 50 seconds. The big advantage of dating schemes is that they provide for constant time algorithms for comparing times and use only linear space in the number of items represented.

Time comparisons are reduced to simple numeric comparisons. Date-based representations are only usable, however, in applications where such information is always known, i.e. applications where every event entered has its absolute date identified. There are many applications where this is a reasonable assumption; for instance, databases of transactions on a single machine, say a central machine maintaining banking records. In addition, with absolute dating, we also have information about the duration of time between events (we simply subtract the date of the later event from the date of the earlier one).

## IV. Temporal Semantic based Access Control Model

In this section we introduce our authorization model, Temporal Semantic based Access Control model (TSBAC), which is an extension of the SBAC model. In TSBAC, we extend the basic authorization model in two directions: adding authorization validation time interval, and associating a temporal expression over a *History Base* (history of users' accesses).

### A. Temporal Authorization Rules with Real Time Scheme

In TSBAC we consider a temporal constraint to be associated with each authorization. This constraint is based on the privileges granted to subjects of the system (on objects), or access requests denied, in a specific real *time point* in the past. These elements of history are stored in *History Base*, in the form of donet,s,o,a and deniedt,s,o,a. We refer to an authorization, together with a temporal constraint and a validation time interval, as a temporal authorization rule. A temporal authorization rule is defined as follows.

**Definition (Temporal Authorization Rule):** A temporal authorization rule is a triple ($[t_s,t_f]$,(s,o,±a),F), where $t_s$∈real-time-sceheme, $t_f$∈real-time-scheme, and $t_s \le t_f$. In this notation, $[t_s,t_f]$ represents the authorization validation time interval, and formula *F* is a temporal constraint which is formally defined as in Table 1.

TABLE 1.
DEFINITION OF TEMPORAL PREDICATE *F*

$$\begin{gathered} A ::= done(s,o,a) | denied(s,o,a) | \\ \sim done(s,o,a) | \sim denied(s,o,a) \\ E ::= prev(A) | past\#(A) | H(A, chunk) | sb\#(A,A) | \\ ab(A,A) | ss(A, chunk) | during(A,A) \\ F ::= true | false | E | {\sim} E | E \wedge E | E \vee E | E \rightarrow E | E \leftrightarrow E \end{gathered}$$

Temporal authorization rule ($[t_s,t_f]$,(s,o,±a),F) states that subject *s* is allowed (or not allowed) to exercise access *a* on object *o* in the interval $[t_s,t_f]$, including time instants $t_s$ and $t_f$, in the case that *F* is evaluated to true.

**Definition (Temporal Authorization Base):** A temporal authorization base (TAB) is a set of temporal authorization rules in the form of ($[t_s,t_f]$,(s,o,±a),F), where $t_s$∈real-time and $t_f$∈real-time.

**Definition (History Base):** A History Base is a set of authorizations and time points, in the form of done(t,s,o,a) which means access *a* has been granted to subject *s* on object *o* at real time point *t,* and denied(t,s,o,a) which means the system has denied access *a* on object *o* at real time point *t* requested by subject *s*.

### B. Informal Meaning of Temporal Authorization Rules

The intuitive meaning of temporal authorization rules is as follows. In these statements *auth* is representative of (s,o, ±a).

- ($[t_s,t_f]$,auth,done(s,o,a)): Authorization *auth* is valid in all time instants *t*, in interval $[t_s,t_f]$, in which done(s,o,a) is evaluated to true. In other words, *auth* is valid at time *t*, if done(t,s,o,a) exists in HB.
- ($[t_s,t_f]$,auth,denied(s,o,a)): Authorization *auth* is valid in all time instants *t*, in interval $[t_s,t_f]$, in which denied(s,o,a) is evaluated to true. In other words, *auth* is valid at time *t*, if denied(t,s,o,a) exists in HB.
- ($[t_s,t_f]$,auth,~done(s,o,a)): Authorization *auth* is valid in all time instants *t*, in interval $[t_s,t_f]$, in which done(s,o,a) is not evaluated to true.
- ($[t_s,t_f]$,auth,~denied(s,o,a)): Authorization *auth* is valid in all time instants *t*, in interval $[t_s,t_f]$, in which denied(s,o,a) is not evaluated to true.
- ($[t_s,t_f]$,auth,prev(A)): Authorization *auth* is valid at the time of request (*t*) in interval $[t_s,t_f]$, if *A* is evaluated to true at the previous moment (t-1). The previous *time point* is determined due to the precision of selected time scheme. For example, if the precision of time is "seconds", the tuple to represent time is of the form of (yyyy dd hh mm ss), so the previous time point is (yyyy dd hh mm (ss-1)). In short, to calculate the previous time point, we simply subtract the numerical representation of time by one. So, PrvTimePoint(yyyy dd hh mm ss)=(yyyy dd hh mm ss)-1 Figure 1 gives a more comprehensible view of the operation of this operator.

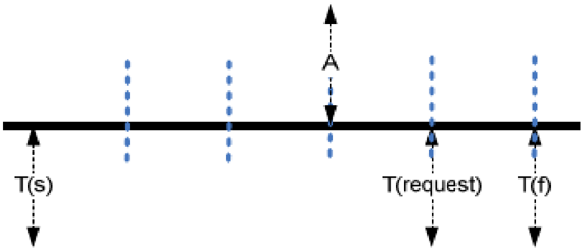


Figure 1. Operation of the *prev* operator on real time axis.

- ($[t_s,t_f]$,auth,past#(A)): Authorization *auth* is valid at the time of request (*t*) in interval $[t_s,t_f]$ if *F* is evaluated to true for # of times from $t_s$ till *t*. 2 gives a more comprehensible view of the operation of this operator.

**Example 1.** Subject $s_1$ is allowed to get another loan (on $Deposit_1$), if he has paid all his (past 36) payments. It is assumed that the date of getting the first loan is 2004/07/01. This rule can be expressed as:

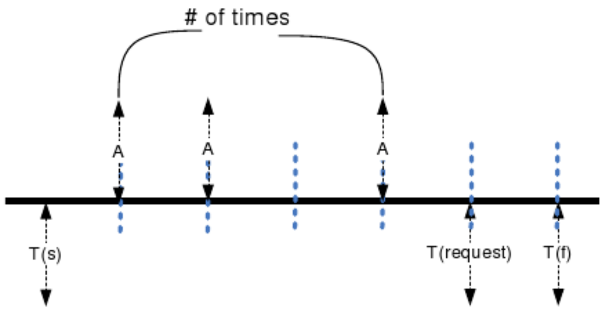


Figure 2. Operation of *past#* operator on real time axis.

$$R:\begin{pmatrix}[2004\ 7\ 1,\infty],(s_1,Deposit_1,+getLoan),\\ past36\big(done(s_1,Deposit_1,payment)\big)\end{pmatrix}.$$

■

- $([t_s,t_f],auth,H(A,chunk))$: Authorization *auth* is valid at the time of request (*t*) in interval $[t_s,t_f]$, if *F* is evaluated to true, at least once in each time interval of length *chunk*, from $t_s$ till *t*. *chunk* is used to reduce the precision of the operator and relax its operation. Figure 3 gives a more comprehensible view of this operator operation.

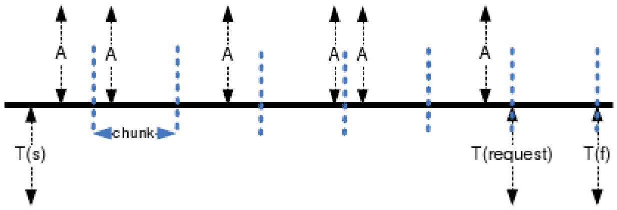


Figure 3. Operation of *H* operator on real time axis.

**Example 2.** Due to insurance rules, everybody can be insured, if he pays for it monthly. This rule for subject $s_1$ that has entered the system since January, 2005 is expressed as: (note that it is assumed a month to be 30 days)

$$R:\begin{pmatrix}[2005\ 1,\infty],(s_1,specialIns,+takeAdvantage),\\ H\big(done(s_1,insDeposit_1,settlement),30\big)\end{pmatrix}.$$

■

All of the operators studied so far, has only one element as their argument. It means that they make their decision (to grant or deny a request) based on presence or absence of just one element in the history base (or first order logic combination of them, but not the temporal relation between two or more of them). In some applications, we need to decide based on the relation between elements of HB. So, TSBAC uses operators that consider the temporal relation between two elements of the history base.

- $([t_s,t_f],auth,sb\#(A_1,A_2))$: Authorization *auth* is valid at the time of request (*t*) in interval $[t_s,t_f]$ if $A_1$ is evaluated to true # of times before the last occurrence of $A_2$, from $t_s$ till *t*. 4 gives a more comprehensible view of the operation of this operator.

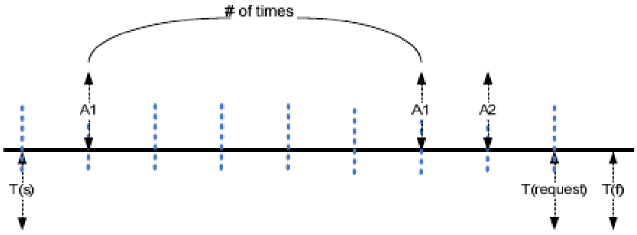


Figure 4. Operation of the *sb#* operator on real time axis

■

- $([t_s,t_f],auth,ab(A_1,A_2))$: Authorization *auth* is valid at the time of request (*t*) in the interval $[t_s,t_f]$ if, if $A_1$ is evaluated to true in $t'\,t_s\le t'\le t$, then there exist a time point $t''$ $t'\le t''\le t$, in which $A_2$ is evaluated to true. In the other words, $A_1$ is evaluated to true in time instants before the evaluation of $A_2$ to true, from $t_s$ till the time of request (*t*). Figure 6 gives a more comprehensible view of the operation of this operator.

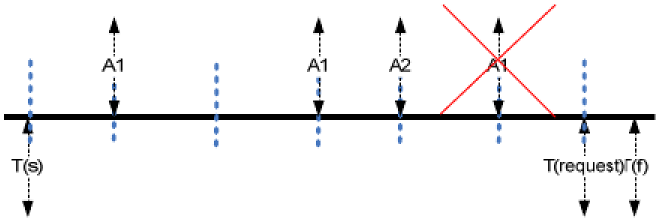


Figure 5. Operation of the *ab* operator on real time axis

**Example 3.** Subject $s_1$ is allowed to get loan on his account (*Account₁*), if he has not withdrawn money from his account since applying for it (that is 2007/01/20). This rule can be expressed as:

$$\begin{pmatrix}[2007\ 20,\infty],(s_1Account_1,+getLoan),\\ ab\begin{pmatrix}\sim done(s_1Account_1,withdraw),\\ done(s_1Account_1,applyForLoan)\end{pmatrix}\end{pmatrix}.$$

■

- $([t_s,t_f],auth,ss(A_1,A_2,chunk))$: Authorization *auth* is valid at the time of request (*t*) in interval $[t_s,t_f]$, if $A_1$ is evaluated to true, at least one time in all time intervals of length chunk, from the first occurrence of $A_2$ in interval $[t_s,t]$. Figure 6 gives a more comprehensible view of the operation of this operator.

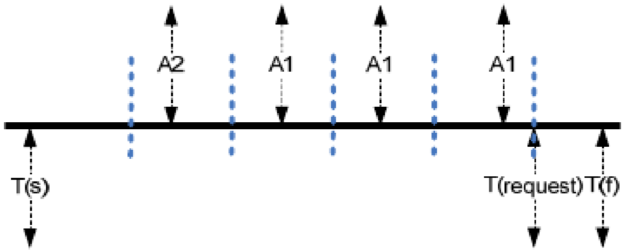


Figure 6. Operation of *ss* operator on real time axis

**Example 4.** Subject s₁ is on the car waiting list, if he paid a prepayment (2006/02/01), and since then he has been paying a defined payment monthly. This rule can be expressed as:

$$\begin{pmatrix} [2006\,31, \infty], (s_1, carWaitingList, +get), \\ ss\begin{pmatrix} done(s_1, Account_1, payment), \\ done(s_1, Account_1, prePayment), 30 \end{pmatrix} \end{pmatrix}.$$

■

- ([t$_s$,t$_f$],auth,during(A$_1$,A$_2$)): Authorization *auth* is valid at the time of request (*t*) in interval [t$_s$,t$_f$] if $A_1$ is not true before the first, or after the last time instant in which $A_2$ is true. Figure 7 gives a more comprehensible view of the operation of this operator.

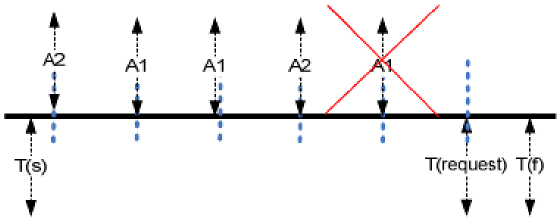


Figure 7. Operation of *during* operator on real time axis

■

- ([t$_s$,t$_f$],auth,~E): Authorization *auth* is valid for each time instant *t* in interval [t$_s$,t$_f$] in which *E* is *not* evaluated to true.
- ([t$_s$,t$_f$],auth,E$_1$∧E$_2$): Authorization *auth* is valid for each time instant *t* in the interval [t$_s$,t$_f$] in which $E_1$ and $E_2$ are both evaluated to true.
- ([t$_s$,t$_f$],auth,E$_1$∨E$_2$): Authorization *auth* is valid for each time instant *t* in the interval [t$_s$,t$_f$] in which $E_1$ or $E_2$ or both of them are evaluated to true.
- ([t$_s$,t$_f$],auth,E$_1$→E$_2$): Authorization *auth* is valid for each time instant *t* in the interval [t$_s$,t$_f$] in which, if $A_1$ is evaluated to true, then $A_2$ is also evaluated to true.
- ([t$_s$,t$_f$],auth,E$_1$↔E$_2$): Authorization *auth* is valid for each time instant *t* in the interval [t$_s$,t$_f$], in which $A_1$ is evaluated to true if and only if $A_2$ is evaluated to true.

### *C. Formal Semantics of Temporal Authorization Rules*

Next we formalize the semantics of authorization rules described so far.

**Definition (Valid Authorization):** an authorization (s,o,±a) is valid at time *t*, if one of the following situations occurred:

1. At time *t*, a temporal authorization rule ([t$_s$,t$_f$],(s,o,±a),F) with t$_s$≤t≤t$_f$ exists in TAB and *F* is evaluated to true based on the elements exist in *History Base* (we define function *f* for performing such an evaluation),
2. There exists a temporal authorization rule ([t$_s$,t$_f$],(s',o',±a'),F) in TAB with t$_s$≤t≤t$_f$ in which *F* is evaluated to true, and (s',o',±a') is derived from (s,o,±a) following the inference rules of SBAC.

- To formalize the semantics of temporal authorization rules, we first define an evaluation function $f_{real}$. This function evaluates the predicate *F* of temporal authorization rules at a real time point *t* and based on the elements stored in *History Base*. The semantics of such an evaluation is given in first order logic and is reported in Table 2. The semantics of a set *X* of temporal authorization rules, denoted by *S(X)*, is the conjunction of the first order formulas corresponding to each element in the set.
- Note that a temporal authorization rule can be removed and therefore not be applicable anymore for the derivation of authorizations. In the formalization we take this possibility into account, by associating with each temporal authorization rule the time $t_d$ at which it is removed. Note that time $t_d$ is not a constant and it is not known from the former. We use it as shorthand for expressing the point up to which a temporal authorization rule is applicable. A function *removed()* can be defined, which, given a temporal authorization rule, *X*, and a time *t* returns *false* if at time *t*, *X* is still present in the TAB, and ,*true*, otherwise. Time $t_d$ is the smallest time *t* for which function *removed(t , X)* returns *true*.

TABLE 2.
FORMAL SEMANTICS OF THE $F_{REAL}$ EVALUATION FUNCTION

$$f_{real}\left(t, t_s, t_f, done(s,o,a)\right) = \begin{cases} true\,, t \in [t_s, t_f] \wedge done(t,s,o,a) \in HB \\ false\,, t \notin [t_s, t_f] \vee done(t,s,o,a) \notin HB \end{cases}$$

$$f_{real}\left(t, t_s, t_f, denied(s,o,a)\right) = \begin{cases} true\,, t \in [t_s, t_f] \wedge denied(t,s,o,a) \in HB \\ false\,, t \notin [t_s, t_f] \vee denied(t,s,o,a) \notin HB \end{cases}$$

$$f_{real}\left(t, t_s, t_f, \sim done(s,o,a)\right) = \begin{cases} true\,, \nexists t, t \in [t_s, t_f] \cdot done(t,s,o,a) \in HB \\ false\,, \exists t, t \in [t_s, t_f] \cdot done(t,s,o,a) \in HB \end{cases}$$

$$f_{real}\left(t, t_s, t_f, \sim denied(s,o,a)\right) = \begin{cases} true\,, \nexists t, t \in [t_s, t_f] \cdot denied(t,s,o,a) \in HB \\ false\,, \exists t, t \in [t_s, t_f] \cdot denied(t,s,o,a) \in HB \end{cases}$$

$$f_{real}\left(t, t_s, t_f, prev(A)\right) = f\left(t-1, t_s, t_f, A\right)$$

$$f_{real}\left(t, t_s, t_f, past\#(A)\right) = \exists t_1, \dots, t_\# \cdot \bigwedge_{k=1}^{\#} f\left(t_k, t_s, t_f, A\right)$$

$$f_{real}\left(t, t_s, t_f, H(A, chunk)\right) = \exists t_0, \dots, t_{\lfloor (t-t_s)/_{chunk} \rfloor - 1} \leq t \cdot \bigwedge_{k=0}^{\lfloor (t-t_s)/_{chunk} \rfloor - 1} f(t_k, t_s + i \times chunk, t_s + (i+1) \times chunk, A)$$

$$f_{real}\left(t, t_s, t_f, sb\#(A_1, A_2)\right) = \exists t_{22} \leq t, \exists t_1, \dots, t_\# \leq t_{22} \cdot f\left(t_{22}, t_s, t_f, A_2\right) \wedge \bigwedge_{k=1}^{\#} f\left(t_k, t_s, t_f, A_1\right)$$

$$
\begin{aligned}
&f_{real}\left(t,t_s,t_f,ab(A_1,A_2)\right)\\
&\qquad=\left(\exists t_1,t_1\le t\cdot f\left(t_1,t_s,t_f,A_1\right)\right)\\
&\qquad\to\left(\exists t_2,t_1\le t_2\le t\cdot f\left(t_2,t_s,t_f,A_2\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
&f_{real}\left(t,t_s,t_f,ss(A_1,A_2,chunk)\right)\\
&\qquad=\exists t_0,\dots,t_{\left\lfloor {(t-t\prime)}/{chunk}\right\rfloor-1}\\
&\qquad\le t\cdot f\left(t',t_s,t_f,A_2\right)\\
&\qquad\wedge\bigwedge_{k=0}^{\left\lfloor {(t-t\prime)}/{chunk}\right\rfloor-1} f(t_k,t'+i\times chunk,t'\\
&\qquad+(i+1)\times chunk,A_1)
\end{aligned}
$$

$$
\begin{aligned}
&f_{real}\left(t,t_s,t_f,during(A_1,A_2)\right)\\
&\qquad=\Big(\Big(\exists t_{min},t_{min}\\
&\qquad\le t\wedge f\left(t_{min},t_s,t_f,A_2\right)\\
&\qquad\wedge\left(\nexists t_x,t_x\le t_{min}\wedge f\left(t_x,t_s,t_f,A_2\right)\right)\Big)\\
&\qquad\wedge\Big(\exists t_{max},t_{max}\\
&\qquad\le t\wedge f\left(t_{max},t_s,t_f,A_2\right)\\
&\qquad\wedge\left(\nexists t_y,t_{max}\le t_y\wedge f\left(t_y,t_s,t_f,A_2\right)\right)\Big)\Big)\\
&\qquad\to\Big(\forall t_1,t_1\le t\\
&\qquad\to\Big(f\left(t_1,t_s,t_f,A_1\right)\to t_{min}\le t_1\\
&\qquad\le t_{max}\Big)\Big)
\end{aligned}
$$

$$
\begin{aligned}
f_{real}\left(t,t_s,t_f,\sim E\right)&=\sim f\left(t,t_s,t_f,E\right)\\
f_{real}\left(t,t_s,t_f,E_1\wedge E_2\right)&=f\left(t,t_s,t_f,E_1\right)\wedge f\left(t,t_s,t_f,E_2\right)\\
f_{real}\left(t,t_s,t_f,E_1\vee E_2\right)&=f\left(t,t_s,t_f,E_1\right)\vee f\left(t,t_s,t_f,E_2\right)
\end{aligned}
$$

$$
\begin{aligned}
&f_{real}\left(t,t_s,t_f,E_1\to E_2\right)\\
&\qquad=f\left(t,t_s,t_f,E_1\right)\wedge\sim f\left(t,t_s,t_f,E_2\right)\\
&f_{real}\left(t,t_s,t_f,E_1\leftrightarrow E_2\right)\\
&\qquad=\left(f\left(t,t_s,t_f,E_1\right)\to f\left(t,t_s,t_f,E_2\right)\right)\\
&\qquad\wedge\left(f\left(t,t_s,t_f,E_2\right)\to f\left(t,t_s,t_f,E_1\right)\right)
\end{aligned}
$$

By the definition of evaluation function freal and by the assumption described above, the semantics of authorization rules are in Table 3 . In the following, grantt,s,o,a denotes subject s is granted to exercise action a on object o and analogously denyt,s,o,a denotes the access request of s for exercising an access a on object o is denied.

TABLE 3.
SEMANTICS OF REAL TIME AUTHORIZATION RULES

$$
\begin{aligned}
&\left(\left[t_s,t_f\right],(s,o,+a),F\right)\\
&\qquad\Leftrightarrow\forall t\left(t_s\le t\right.\\
&\qquad\left.\le min\left(t_f,t_d-1\right)\wedge f_{real}\left(t,t_s,t_f,F\right)\right)\\
&\qquad\to grant\left(t,(s,o,a)\right)
\end{aligned}
$$

$$
\begin{aligned}
&\left(\left[t_s,t_f\right],(s,o,-a),F\right)\\
&\qquad\Leftrightarrow\forall t\left(t_s\le t\right.\\
&\qquad\left.\le min\left(t_f,t_d-1\right)\wedge f_{real}\left(t,t_s,t_f,F\right)\right)\\
&\qquad\to deny\left(t,(s,o,a)\right)
\end{aligned}
$$

### D. Access Control

The centric security mechanism in each system is an access control system. By receiving an access request in such a system, we need to make a decision whether to grant the requested access or deny it. Following the proposed model of temporal authorization in the previous sections, upon receiving an access request sr,or.ar at time *t*, the access control system performs the following steps:

1. Determine the explicit and implicit valid authorization rules in TAB at time *t* (following the definition of valid authorization rules), satisfying the following conditions:
   - $t_s \le t \le min(t_f,t_d)$
   - Temporal predicate *F* is evaluated to true at time *t* (based $f_{real}$ evaluation function).
2. Extract the set of valid authorization rules such as ([$t_s$,$t_f$],(s,o,±a),F) which match the access request. These authorization rules must satisfy, at least, one of the following conditions:
   - s=$s_r$ , o=$o_r$ , a=$a_r$
   - Following the propagation rules of the SBAC model, in the case of a positive action (+a), we have sr<s , or<o , ar<a, and in the case of a negative action (-a), we have sr<s , or<o , a<ar.
3. If there exist just positive valid authorization rule(s) such as ([$t_s$,$t_f$],(s,o,+a),F) in MVA, grant the requested access,
4. If there exist just negative valid authorization rule(s) such as ([$t_s$,$t_f$],(s,o,-a),F) in MVA, deny the access request,
5. If there exist both positive and negative authorization rules in MVA, do conflict resolution and follow the result,
6. If there exists no valid authorization rule, which matches the requested access, follow the default access policy,
7. Record done($s_r$,$o_r$.$a_r$) in the case that the requested access is granted, and denied($s_r$,$o_r$,$a_r$) in the case that the access request is denied.

In this model, the default access policy might be *positive (open)* to grant all undetermined accesses, or *negative (close)* to deny them. The default access policy is determined by the administrator.

### E. Conflict Detection and Resolution

A conflict occurs when two or more access policies cannot be applied in the same time. In access control, due to modal conflict between *matched valid authorizations*, we need a conflict resolution strategy.

#### 1) Conflict Occurrence

In TSBAC, conflict occurs due to semantic relations between entities (in the domains of subjects, objects, or actions) and applying the inference rules of SBAC, or due to sub-interval relationship between the temporal authorization rules of the TSBAC model.

- *Conflict due to semantic relations between the entities*: as mentioned before, in the domains of subjects and objects, the subsumee has all the privileges (positive and negative) of the subsumer, but, in the domain of actions, positive access rights is propagated from subsumer to subsumee, while negative access rights is propagated in the opposite direction (that is from subsumee to subsumer). These semantic relationships and the propagation of negative and positive authorizations between the entities may result in conflicting situations. As an example, consider the following History Base, semantic relation, and authorization rules:

$$\begin{array}{l}
\boldsymbol{HB\ Contains}: \\
done(10, Student, doc_1, read), \\
done(8, Ali, doc_1, write) \\
R_1: \begin{pmatrix} [0,25], (Student, doc_1, +read), \\ prev\big(done(Student, doc_1, read)\big) \end{pmatrix} \\
R_2: \begin{pmatrix} [0,25], (Ali, doc_1, -read), \\ prev\big(done(Student, doc_1, read)\big) \end{pmatrix} \\
\boldsymbol{Subjects\ Ontology}: \\
Ali \prec Student
\end{array}$$

If *Ali* requests a *read* access at time *11* , due to $R_2$ authorization rule, this access is denied, but due to the $R_1$ authorization rule we have Student,doc1,+read , and based on the sample subjects ontology, the *read* access is also granted for *Ali* . So, in this situation we are confronted with a conflicting situation, due to the semantic relationships between the entities.

- *Conflict due to sub-interval relationship between authorization rules*: as an example, consider the following HB, and authorization rules:

$$\begin{array}{l}
\boldsymbol{HB\ Contains}: \\
done(8, Ali, doc_1, read), done(9, Ali, doc_1, read), \\
done(10, Ali, doc_1, read) \\
R_1: \begin{pmatrix} [8,20], (Ali, doc_1, +read), \\ H\big(done(Ali, doc_1, read)\big) \end{pmatrix} \\
R_2: \begin{pmatrix} [0,20], (Ali, doc_1, -read), \\ prev\big(done(Ali, doc_1, read)\big) \end{pmatrix}
\end{array}$$

If *Ali* requests a *read* access at time *11* , due to $R_2$ authorization rule, this access is denied, but is granted due to the $R_1$ authorization rule. So, we are confronted with a conflicting situation based on the sub-interval relationship between $R_1$ and $R_2$.

#### 2) Conflict Resolution

The model supports four predefined strategies for conflict resolution; negative authorization rule takes precedence (NTP) strategy, positive authorization rule takes precedence (PTP) strategy, most specific authorization rule takes precedence, and newer authorization rule takes precedence. Similar to default access policy, the conflict resolution strategy is determined by the administrator.

### F. Temporal Authorization Base Administration

Authorization rules can be changed upon the execution of administrative operations. In this paper, we consider a centralized policy for administration of authorizations where administrative operations can be executed only by the administrator.

Administrative operations allow the administrator to add, remove, or modify (a *remove* operation followed by an *add* operation) temporal authorizations rules. Each temporal authorization rule in the TAB is identified by a unique label assigned by the system at the time of its insertion. The label allows the administrator to refer to a specific temporal authorization rule upon execution of administrative operations. A brief description of the administrative operations is as follows:

*addRule*: To add a new temporal authorization rule. When a new rule is inserted, a label (*rule identifier* or *rid*) is assigned by the system.

*dropRule*: To drop an existing temporal authorization rule. The operation requires as argument, the label of the rule to be removed.

## V. Architecture

In order to guarantee the applicability of the model and usefulness in semantic based and temporal environments, an architecture for the temporal semantic based access control model is proposed.

Several frameworks has been proposed for access control and security in recent years, which, the standard framework of the ITU-T [17] for access control and the standard framework OASIS under the name XACML [18] are the most popular ones. during the design of our model, we have tried to include the elements of both the frameworks mentioned above, especially, the XACML. The major elements of the system are illustrated in .8 The system is composed of a number of *internal* and *external* elements which are described next.

### A. External Entities

The major external entities interacting the system are:

***Subject***: A subject can be a person, a service, or a machine that tries to access resources or objects in a semantic based environment.

***Environment***: The set of *attributes* that are relevant to an *authorization decision* and are independent of a particular *subject, resource*, or *action*.

***Objects and access rights***: entities which provide ontologies in the domains of subject, object, and action. These ontologies and the semantic relations between them are helpful in propagation and inference of new security rules.

### B. Administration Console

This console enables the security administrator to describe meta-policies and also description and administration of the ontologies. The major components of the administration console are as follows:

***Policy Administration Point (PAP)***: The system entity that creates a policy or policy set.

***Ontology manager***: Gathers and updates ontologies in domains of subjects, objects, and actions and also reduces the semantic relations to the subsumption relation.

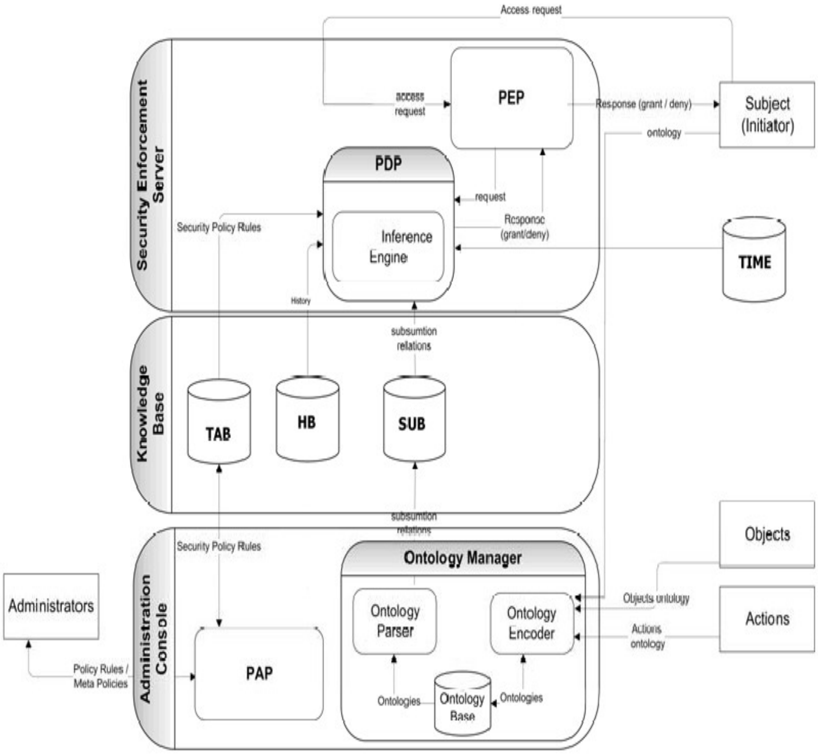


Figure 8. An architecture for an access control system based on TSBAC model

### C. Knowledge Base

As described in the model, the knowledge base is composed of the set of *authorization rules* in TAB, *history of authorizations* (HB), *subsumption relations* between concepts (SUB), and a *time counter* (TIME). Inference of implicit authorization rules is based upon the facts and rules in the knowledge base.

### D. Security Enforcement Server

This element manages inference of the implicit authorization rules and applying them in access control requests. Major sub-elements of this element are as follows:

***Policy Decision Point (PDP)***: The system entity that evaluates *applicable policy* and renders an *authorization decision* by making use of an inference engine, based on facts and rules in the knowledge base.

***Policy Enforcement Point (PEP)***: The system entity that performs access control, by making decision requests and enforcing authorization decisions.

## VI. Discussion and Evaluation

Evaluating authorization models and access control mechanisms, and presenting acceptable criteria in this domain, has been a problem in security and access control zone. Comparing security models with each other, due to differences between them in security definition, seems improper. Security is a comparative quality, and assumptions in security definitions in an environment and security requirements in that environment makes distinctive differences in designing the model. The best way to evaluate a model is to qualitatively scrutiny the model to ensure accordance with security requirements of environment under custody. Moreover, we can take some quantitative criteria into account, but this consideration is possible if an implementation exists for the access control system based on the model.

### A. Qualitative Evaluation of TSBAC Model

In this section, we evaluate TSBAC, regarding requirements of semantic and history based environments.

- Fine-grained and Coarse-grained Authorization: TSBAC allows definition of policies for entities in three domains of access control (namely subjects, objects, and actions), so it provides coarse-grained authorization. Moreover, with the existence of ontology, and possibility of defining entities to the individual level, fine-grained authorization is provided.
- Conditional Authorization: With the existence of temporal operators, TSBAC supports this type of authorization. In this model, due to wide spectrum of temporal operators, and using first order logic operators for combining temporal expressions, conditional authorization is provided, on the basis of existence or nonexistence of specific authorizations in the past.
- Different Policies and Expressing Exceptions: TSBAC provides synthetic policy (including negative and positive authorizations). Moreover, by using ontology in domains of subjects, objects, and actions, and utilizing different authorization propagation methods, expressing exceptions and synthetic policies is possible.

- Conflict Detection and Resolution: Conflict occurrence may be a result of semantic relationships between authorizations, or, sub-interval relations between validity constraint intervals. TSBAC detects these conflicts, and resolves them. Different conflict resolution policies include: denials take precedence, positives take precedence, most specific takes precedence, and newer overrides older.
- Ease of Implementation and Integration with Semantic Web technologies: Security models designed for Semantic Web should be compatible with the technology infrastructure under it. In other words, the implementation of security mechanisms should be possible based on the semantic expression models. SBAC is designed based on the widely accepted semantic web languages, OWL and SWRL; therefore its implementation can be easily achieved by existing tools designed for working with these languages.
- Supporting History-based Information: The main feature of TSBAC is that authorizing an access request is done based on granted or denied access requests (done and denied access requests, which are stored in History Base), or, access requests that have not been done or have not been denied in the system (which can be inferred from History Base). These elements could be combined with temporal operators, or first order logics operators to compose temporal expressions.
- Interoperability : Interoperating across administrative boundaries is achieved through exchanging authorizations for distributing and assembling authorization rules. The ontological modeling of
- Authorization rules in SBAC results in a higher degree of interoperability compared with other approaches to access control. This is because of the nature of ontologies in providing semantic interoperability.
- Generality: Modeling different domains of access control has added a considerable generality to the model. In the subject domain, TSBAC uses credentials which are going to be universally used for user authentication. In the domain of object, different kinds of resources such as web pages or web services can be modeled and can be identified by their URI in authorization rules.

### B. *Quantitative Evaluation of TSBAC*

- Time Complexity: Since every access request is validated at the time of the request, and the process of authorization is based upon searching History Base and evaluating the temporal predicate, due to vast amount of elements of History Base and temporal predicate complexity, access control in TSBAC is time consuming. In some situations, in order to evaluate the temporal predicate, we need to scrutinize the existence of "not-done" or "not denied" requests, and this adds to the time complexity of access control process.

In order to clarify the subjects mentioned above, we give a brief complexity analysis on real time operators (in case of existence of *n* elements in History Base) of the model:

complexityprev=n<br>
complexityH=t-ts/chunk×n<br>
complexitypast#=#×n<br>
complexitysb#=#+1n<br>
complexityab=n<br>
complexityss≤t-ts/chunk×n<br>
complexityduring=n

- Space Complexity: All of the access requests (granted or denied) are stored in History Base. Storing all the requested accesses in the system, gradually, requires a huge amount of storage space. In case of a vast amount of history elements, and thus incapability of keeping all these elements on volatile storage, time complexity of access control process is amplified.

## VII. Conclusions and Future Work

Access control and its requirements in new computing environments, semantic aware access control, and history based access control have been discussed in this paper. Based on the Semantic Based Access Control model (SBAC), and to enhance the capabilities of the model, a semantic aware access control model, which takes the history of accesses of the system into account (TSBAC) is proposed. TSBAC uses the same semantic relationships of the SBAC model, and moreover, it is capable of using temporal relations between authorizations in applying access control. Specifically, TSBAC assigns a temporal expression (over users' history of accesses) to each authorization that expresses the conditions under which the authorization applies. A constraining time interval restricts the interval of validity of the authorization. These authorization rules (which are composed of base authorization of SBAC, constraining time interval, and temporal expression), provides the ability to derive new authorizations based on existence (or non-existence) of other authorizations in the past.

We also proposed formal semantics of our authorization rules. Access control, and conflict detection and resolution presented. An architecture for the access control system based on TSBAC was presented.

Producing preconditions of applying temporal logics operators in Java language, and using these preconditions in CLIPS inference engine in order to apply access control can be considered as some future works.

One of the main deficiencies of TSBAC is the lack of a formal proof for soundness and completeness of temporal operators of the model. On the other hand, a generalized history-based access control model that could be applied to other access control policies (such as RBAC) is one of the important works that could be done.

## References

[1]. Samarati, P. and S.C.d. Vimercati, *Access control: Policies, models, and mechanisms.* Foundations of Security Analysis and Design, LNCS, 2001. **2171**: p. 137-196.

[2]. S. Javanmardi, A. Amini, and R. Jalili. *An Access Control Model for Protecting Semantic Web Resources*. in *Web Policy Workshop*. 2006. Ahens, GA, USA.

[3]. Javanmardi, S., et al. *SBAC: "A Semantic-Based Access Control Model"*. in *NORDSEC-2006*. 2006.

[4]. Bonatti, P.A., et al., *Semantic web policies: a discussion of requirements and research issues*. ESWC, 2006. **2006**: p. 712-724.

[5]. RABITTI, F., et al., *A Model of Authorization for Next-Generation Database Systems*. ACM TODS, 1991. **16**(1): p. 87-99.

[6]. Qin, L. and V. Atluri. *Concept-level access control for the Semantic Web*. in *2003 ACM workshop on XML security*. 2003.

[7]. Brewer, D.F.C. and M.J. Nash. *The Chinese Wall Security Policy*. in *IEEE Symposium on Security and Privacy*. 1989. Oakland, California.

[8]. Dias;, P., C. Ribeiro;, and P. Ferreira, *Enforcing History-Based Security Policies in Mobile Agent Systems*. 2003.

[9]. Edjlali, G., A. Acharya, and V. Chaudhary. *History-based access control for mobile code*. in *5th ACM conference on Computer and communications security*. 1998.

[10]. Abadi, M. and C. Fournet. *Access control based on execution history*. in *10th Annual Network and Distributed System Security Symposium*. 2003.

[11]. Bertino, E., et al., *A temporal access control mechanism for Database Systems*. IEEE Trans. Knowl. Data Eng, 1996. **8**(1): p. 67-80.

[12]. Thomas, R.K. and R.S. Sandhu. *Sixteenth National Computer Security Conference*. 1993. Baltimore, Md.

[13]. Bertino, E., C. Bettini, and P. Samarati. *A temporal authorization model*. in *Second ACM Conference on Computer and Communications Security*. 1994. Fairfax, Va.

[14]. Bertino, E., et al., *An access control model supporting periodicity constraints and temporal reasoning*. ACM Trans. Database Systems, 1998. **23**(3): p. 231-285.

[15]. Ruan, C., *Decentralized Temporal Authorization Administration*. 2003.

[16]. Bertino, E., et al., *Temporal authorization bases: From specification to integration*. Journal of Computer Security, 2000. **8**: p. 309-353.

[17]. ISO/IEC:10181-3. *Information Technology - Open Systems Interconnection - Security Frameworks for Open Systems: Access Control Framework*. 1995.

[18]. Moses, T., *eXtensible Access Control Markup Language, Version 2.0*. 2005.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Network-level properties of modern anonymity systems

Marta Rybczyńska
Warsaw University of Technology

Email: mrybczyn@elka.pw.edu.pl

***Abstract*—This paper shows the network-level view of the behaviour of two popular and deployed anonymity systems; Tor and JAP (AN.ON). The analysis uses the fact that both of them depend on TCP (Transmission Control Protocol) and shows cases when network conditions may affect the systems' operations. The main topics are: on-off traffic, difference in available bandwidth between peers, and usage of the window mechanism in the TCP. The analysis is divided into two parts: from both the sender's and receiver's point of view. We present the results of experiments made on live systems, showing that the analysed effects may be encountered in practice.**

## I. Introduction

THIS PAPER studies network-level effects on performance and protection given by modern anonymity systems. As the anonymity layer is built on top of the existing network protocols, the interactions between the two may affect the operations of the anonymity system.

We focus on the effects the TCP protocol has on the two most popular and already deployed designs; Tor and JAP. Although a significant amount of effort has been put into researching possible attacks against the protection mechanisms used by those systems, it is not clear which features of the network protocol allow the system to gain better protection, and which should be avoided.

It has been shown that attacks by traffic analysis against Tor are possible, when the attacker modifies the traffic from the sender's side [21]. We check if this works equally well against both Tor and JAP, and look into the limits of such attacks, introduced by the network conditions. We do not limit the analysis to the sender's side only, but also investigate the way the receiver can shape the traffic using the TCP flow control. The analysis and interpretation of results concentrate on the security-related issues, but we also look into possible performance implications when they are clear of issues.

The remainder of this paper is structured as follows: Section 2 reviews the related work and gives the background on Tor and JAP. In Section 3 we show our methodology and setup. The analysis and results are presented in Section 4. We conclude the paper in Section 5.

## II. Background

Modern anonymity systems can be divided into two types. The first type introduces a noticeable delay, even up to days, in exchange for good anonymity of the user messages. Such systems are designed for services like e-mail. Two important examples are Mixminion [6] and Mixmaster [13]. The second group can be used for real-time or nearly real-time applications, like web browsing or remote access. Such systems are designed using client-server or peer-to-peer model. Important examples include Tor [7], JAP (AN.ON) [2], Freedom [3] (all three client-server), and the new I2P system [8] (peer-to-peer).

### A. Tor

Tor is a low-delay, high-bandwidth anonymity system developed for TCP traffic, like web browsing [7]. It includes several extra features, like hidden servers.

The current implementation uses TCP connections between the nodes. It neither supports cover traffic, nor mixing. Congestion control and simple traffic shaping exist. The flows are routed independently through the network. However, multiple requests transmitted between the same two intermediate nodes may be multiplexed and form one connection.

When it comes to the network layer, Tor uses TCP/IP implementation provided by the host operating system. All the processing is done at the user level. From the networking point of view, Tor is a cascade of proxy servers with traffic aggregation.

### B. JAP (AN.ON)

JAP (Java Anon Proxy) is an anonymity system developed at the University of Dresden [10] [11].

It uses a different approach than Tor. Instead of onion routing, it uses a modified version of Chaum mixes. JAP does not form one single network. Instead, users connect to so called cascades. In a cascade, mixes are connected by single connections only, which are used to transfer all the traffic. Additionally, proxy cache servers may be added after the last mix. In practice, JAP uses static cascades. All flows share the same path and are transmitted in a single TCP connection between subsequent nodes (mixes).

From the networking point of view, JAP is, just like Tor, a cascade of proxy servers also implemented as external applications.

### C. Related work

The influence of traffic on the operations of the anonymity systems has been studied mainly in the context of traffic analysis attacks. That class of attacks uses flow properties to match flows before and after the anonymity system.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

In the literature, a statistical approach is often used, as in [5]. Murdoch and Danezis have shown an attack against Tor that marks traffic flows by introducing delays, caused by flooding a node with traffic from an attacker–owned, corrupted node [14]. On the other hand, Wiangsripanawan et al. state that the attack would not work against systems using a different design, like that of Tarzan [19]. Zhu et al. have used a different approach and studied flow correlation attacks for a mix–based system [22]. They propose dummy traffic as a defence.

The problem of timing attacks have been studied in a number of papers [12] [16] [20], showing that numerous properties of the traffic can be observed at the other end of the anonymity system. Shmatikov et al. propose dummy traffic against such attacks [17]. It has also been shown that single flows can be recognised in the aggregated traffic [23]. Yu et al. introduce a flow marking technique where they mark traffic by the changing transfer rate, and detect the introduced pseudo-noise code later [21].

Research on the performance of current designs and reasons for that performance has been much less intensive. Wendolsky et al. measured the delay introduced by Tor and AN.ON (JAP), and report it to be in seconds, rather than milliseconds [18].

## III. Setup and methodology

Our test environment consisted of two computers, controlled by us, and connected to the Internet. One of the machines had a Web server installed, the other one worked as client. The client also used Tor and JAP client software, which was used to connect to the anonymity systems.

During the test we downloaded several files of different sizes, from 40MB to 100KB, from our server to the client, using the two chosen anonymity systems during their normal operations. We did not control any of the nodes, nor interfere with the standard path selection algorithms.

We introduced changes to the traffic sent by the server and received by the client using bandwidth limits by utilising the standard Linux firewall tool iptables/netfilter with additional, custom scripts. We examined the direction from the sender using traffic shaped into bursts with lengths of 60 and 10 seconds, and with a constant limit during experiments performed on the receiver. There were two types of bursts, with the first at 128 kbit/sec and the second at 64 kbit/s. Between bursts, the transmission took place at 32 kbit/sec. Such rates were chosen after initial experiments, showing that JAP allows us to transfer at a maximum rate of 128 kbit/sec. All the traffic was captured for later analysis.

We used unmodified client software and only services available to the general public. It turned out that while huge a majority of Tor connections offered acceptable performance (there were, however, sporadic connection resets and stops of the data flow), JAP showed the performance needed for the experiments only on one cascade; Desden-Dresden. On the other cascades, unexpected connection drops and/or very low bandwidth prevented even short file transfers.

As we did not control the anonymity systems or the behaviour of other users, we could not achieve full stability of the conditions for our measurements. We addressed that issue at different levels. Our transfers were relatively long (from several seconds up to minutes), what allowed us to limit the impact of short disturbances, which were observed in the traffic. Low transfer rates allowed us to perform transfers without interfering with other limits there may be on the path, and without putting significant stress on the anonymity system nodes. Finally, we performed the experiments during the same hours and days of the week.

The later analysis was performed using the recorded traces. We processed the traces to get the number of packets in a one second timeframe. Then we calculated the cross-correlation between both ends. We used the following equation:

$$r(d) = \frac{\sum_i [(x(i) - mx) * (y(i-d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i-d) - my)^2}} \qquad (1)$$

where $d$ is delay, $r$ is cross–correlation, $x(i)$ and $y(i)$ are the signals, and $i = 0, 1, ..N - 1$. $mx$ and $my$ are the mean values of the $x$ and $y$ signals, respectively. Cross-correlation is often used as a metric in anonymity system evaluation ([22], [12] or [21]), because it gives an attacker a relatively simple and efficient tool to match the sender with the receiver. We used that metric for the same reasons.

## IV. Analysis and results

The analysis is based on TCP properties and on the properties of TCP connection cascades. Similar tests may be performed for any other TCP-based anonymity system.

Figure 1 shows one of our Tor transfers with no modifications from our side, in which a number of interesting facts arise. The first one is the disturbance at approx. 400 seconds from the transfer start, which is clearly visible in both the server and client traffic. Such disturbances can cause attacks to fail and were also observed in the other traces.

It is also worth noting that the number of packets on both pictures differs. The server sends at approx. 20 packets/sec, while the client receives at 40 packets/sec. That fact is easy to explain after realizing that the first picture shows plain a HTTP response, when the second shows the same response, but for packed and encrypted in Tor messages. That is why those pictures, and the ones shown later, should not be compared directly.

Figure 2 shows similar graphs for JAP traffic, with visible bursts on the server side (similar ones appear in all of our transfers) which looks like the expected results of the backward attack. The client traffic does not run at the maximum rate, however, as there are also bursts visible from that side.

### A. Sender side

A number of different modifications to the traffic may be made on the sender side. For instance, the server may limit bandwidth for a single flow. There may also be bursts of traffic as the server switches from one flow to another. Additionally, there may be losses, which cause TCP retransmissions. We

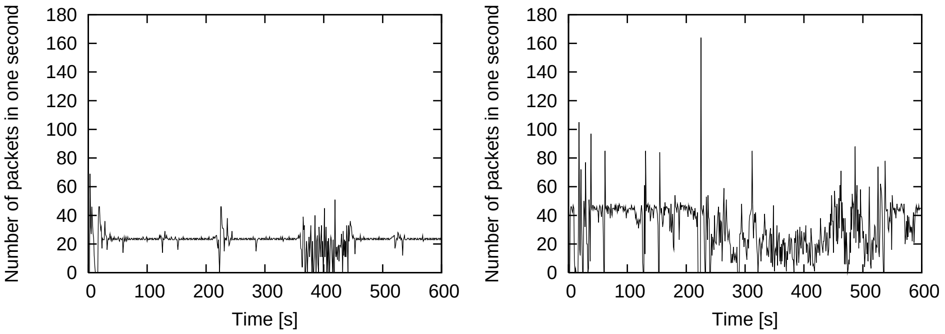


Fig. 1. Traffic flow sent from the server (left) and received by the client (right) during a transfer using Tor, without modifications to the traffic.

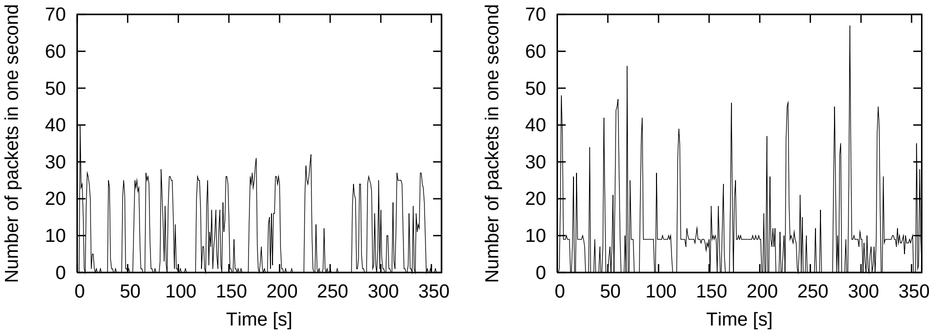


Fig. 2. Traffic flow sent from the server (left) and received by the client (right) during a transfer using JAP, without modifications to the traffic.

wanted to check if and how echos of such events would be visible on the receiver side.

Such echos are likely because of one of the basic features of the TCP protocol [9] [15], namely the fact that the data reaches the receiver application in the same order it was sent, even if it might have reached the node in a different order. It means that if one message is delayed, the data from the later ones would not reach the receiver before the delayed message is received correctly. Given the fact that the path used by the anonymity system consists of several TCP connections, the delays will cumulate.[1] That analysis is correct for both Tor and JAP, as they both use the TCP for transfers between the nodes and between the user and the nodes.

Such properties were already discussed in the context of traffic analysis attacks, as in [14] or [12]. We are not aware, however, of effectiveness tests under real-world conditions.

Figures 3 and 4 show modulated traffic before and after it is transferred, using Tor and JAP, respectively. In both cases the anonymity system did not change the delays and transfer rates to the extent that would make it unrecognisable.

In the tests we introduced a known traffic pattern into the anonymity systems and observed the resulting flow on the receiver side. In the first test, our server was transmitting at 128kbit/sec for 60 seconds, then at 32kbit/sec for 120 seconds, then at 64kbit/sec for 60 seconds and then, finally, at 32kbit/sec again for another 120 seconds. The relatively slow transmission rates were used to limit the possibility of reaching any other limits there might be on the path. The results are presented in Tab. I. Tor traffic obviously shows high correlation. In the case of JAP, the correlation is not that easily seen. However, after calculating cross-correlation between the flows using the equation (1), the values for the corresponding flows are the highest.

TABLE I
CROSS–CORRELATION BETWEEN THE USER AND SERVER SIDE TRAFFIC.

| System | Test | Best correlation |
|---|---|---|
| None | Normal transfer | 0.97 |
| Tor | Long bursts | 0.82 |
| | Short bursts | 0.23 |
| JAP | Long bursts | 0.54 |
| | Short bursts | 0.59 |

TABLE II
CHANGE OF BURST LENGTH AFTER PASSING THROUGH ANONYMITY SYSTEMS (ORIGINAL LENGTH: 60 SAMPLES).

| System | Burst length (before) | | Burst length (after) | |
|---|---|---|---|---|
| | avg. | std. dev. | avg. | std. dev. |
| Tor | 50.0 | 4.1 | 59.4 | 9.3 |
| JAP | 47.9 | 7.0 | 52.7 | 10.1 |

We have also calculated the length of the burst before and after the anonymity system. The results are presented in Table II. In both cases, bursts after the anonymity system are longer

[1] That is independent from packet reordering. The packets will be placed back in the correct order on every anonymity system node during TCP data reassembly, before passing them to the application (anonymity software, in this case).

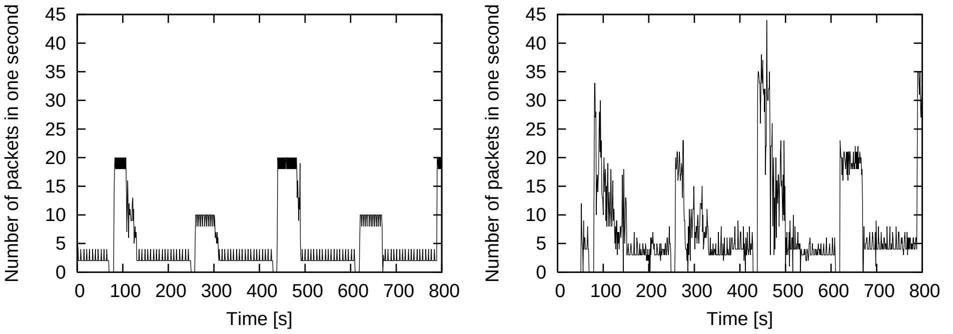


Fig. 3. Traffic flow sent from the server (left) and received by the client (right) during a transfer using Tor, sender–side traffic shaping.

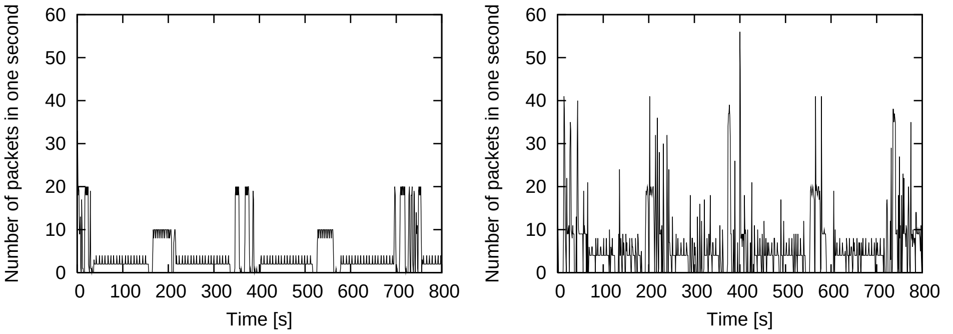


Fig. 4. Traffic flow sent from the server (left) and received by the client (right) during a transfer using JAP, sender–side traffic shaping.

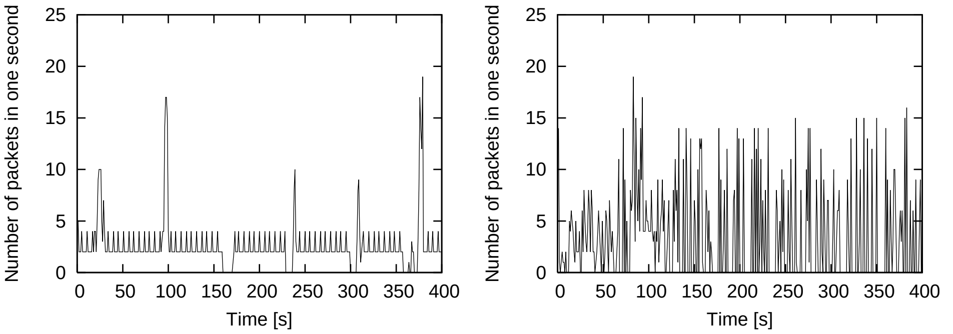


Fig. 5. Traffic flow sent from the server (left) and received by the client (right) during a transfer using Tor, server-side traffic shaping with short bursts.

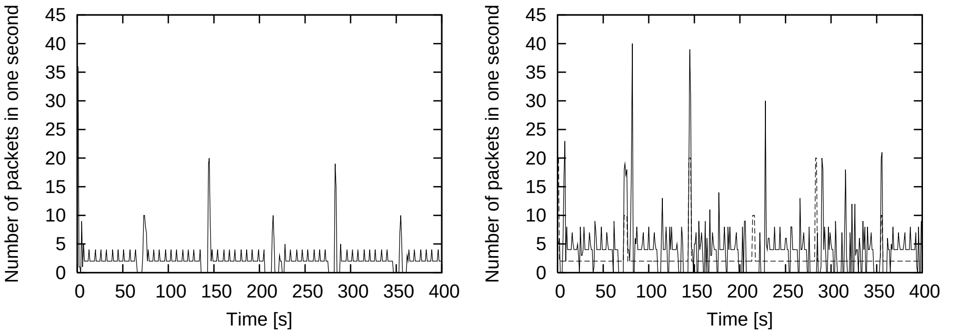


Fig. 6. Traffic flow sent from the server (left) and received by the client (right) during a transfer using JAP, server-side traffic shaping with short bursts.

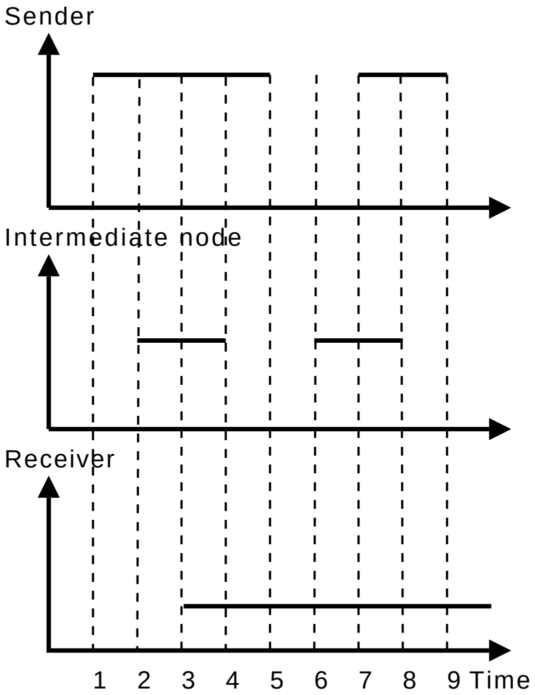


Fig. 7. Application-level view of the traffic in TCP-based anonymity systems as a result of flow control, in the case of differences in available bandwidth, from the sender (S) to the receiver (R) using the intermediate node (IM). Main events: (1) S starts transmitting to IM, (2) IM starts transmitting to R, (3) R starts getting data, (4) IM's transmit buffer full, stops receiving from S, (5) S' transmit buffer full, stops sending, (6) IM re-starts sending to R and receiving from S, (7) S re-starts sending to IM, (8) IM's transmit buffer full again, (9) S' transmit buffer full again.

than before, but the standard deviation also increases.

As the test was successful, we performed another one, using shorter, 10 second bursts. There were two types of bursts: at 128 and 64 kbit/sec. The traffic was then formed at the server, as shown in Figures 6 and 5. It can be noticed that the bursts are actually shorter than 10 seconds, which is an effect of the rate switching.

This time, the Tor traffic shown in Figure 5 did not show similarities to the original. That was also confirmed by low cross-correlation.

The JAP traffic shown in Figure 6, on the other hand, shows some similarities to the original. The cross-correlation test gave values at nearly the same levels as in the previous case of longer bursts.

### B. Receiver side

Conditions on the receiver side can also influence the whole path because of the flow control mechanism in the TCP. The TCP has a window mechanism that allows notifications to the other node, that the transmission should be slowed down or stopped. A window is the number of octets that may be in the network without an acknowledgement. The current value is transmitted as a field in the TCP header [1] [4].

Dividing one TCP connection into multiple ones breaks that end-to-end flow control. The result of this may be seen if the sender transmits at a higher rate than the receiver can receive, and if the rate offered by the anonymity system is enough to handle the traffic without introducing noticeable additional delays or losses.

The situation is depicted in Figure 7. The transmission on the first (receiver) link will take place at the maximum rate (or close to that value). It will be lower, from the beginning, than the transmission rate of the server (events 1-3). That will cause the buffers in the nodes to fill.

When the sending buffer on the first node becomes full, it lowers the window size in the connection with the second node. After some time, the sending buffer on that node will become full. That will stop receiving from the next node (event 4). Finally, window lowering will occur at the last link (between the sender and last node), where it can be observed (event 5). With the maximum rates still at the same level, the transmission from the server will start forming bursts, as the transfers will occur at the maximum rate, but only for short periods of time, when the next node allows it (events 6-9).

Depending on the network configuration, delays, TCP implementation used by the nodes, and their settings, the bursts may be observed sooner or later. Also, the time until the window drops to zero for the first time will vary. In the worst case that includes sending and receiving TCP buffers on all the anonymity system nodes along the path and the buffers in the anonymity software. The bursts may appear earlier if the anonymity protocol has its own window mechanism, for instance.

*Long and short bursts:* We have checked if the effect described above can be observed in practice. Figures 8 and 9 show the results for Tor and JAP, respectively.

In the case of Tor there are bursts visible which are just as we expected (compare the with normal traffic from Figure 1). The time from the transfer start to the moment when the burst appears is much lower than expected, however. That was the reason for another test. The aim was to check more Tor paths and see how long a time was needed for the bursts to appear. We show the results later in this section.

JAP also shows bursts. The client receives at the maximum rate. The bursts do not differ sufficiently from the ones in Figure 2. Even the burst lengths are similar. A different pattern did not emerge even during the longest 12 minutes transfer.

The fact that the test did not work against JAP during our test time, is one of the issues worth noticing. The root cause remains a problem to be considered in the future work. The bursts may finally appear after all the buffers on the path are filled, which may take hours in the case of the slow transfer rates we used.

Our observations show that the transfer rates, even without any modifications, were lower for JAP than Tor. Also, Figure 6 shows no bursts on the server side, that may suggest an external bandwidth limit.

*Filling the buffers of Tor:* We performed additional experiments on Tor, as the amount of data needed to start bursts was much lower than expected. The results we got clearly show that the buffers on the whole path were not full when

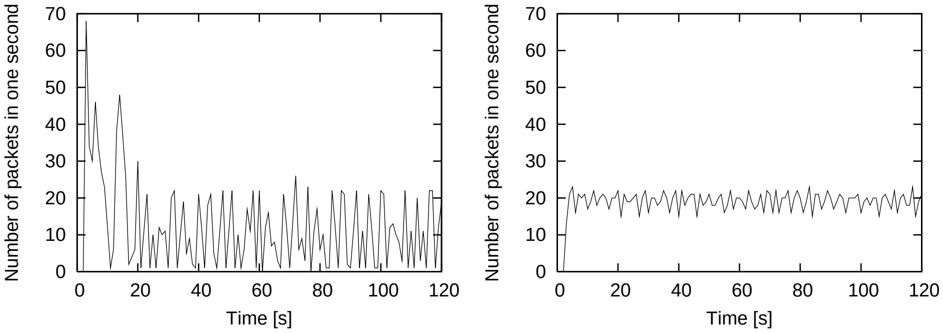


Fig. 8. Traffic flow sent from the server (left) and received by the client (right) during a transfer using Tor with client bandwidth limit at 96kbit/s.

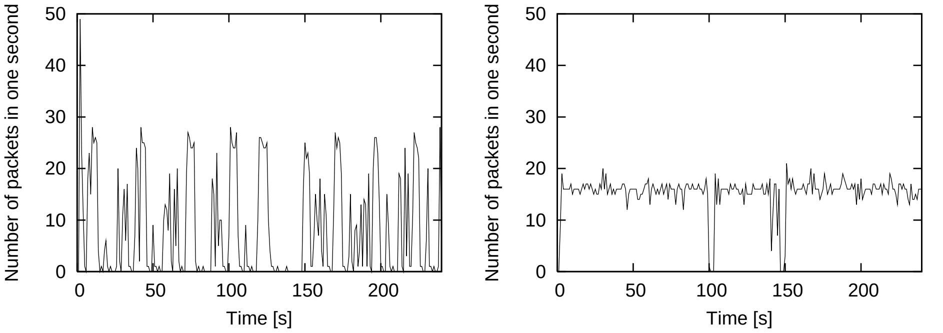


Fig. 9. Traffic flow sent from the the server (left) and received by the client (right) during a transfer using JAP with client bandwidth limit at 96kbit/s.

TABLE III
DELAY TO THE FIRST ZERO WINDOW MESSAGE.

| Amount of bytes transferred | Number of occurrences | Percentage |
|---|---|---|
| less than 100,000 | 4 | 15.4% |
| 100,000 to 200,000 | 2 | 7.7% |
| 200,000 to 300,000 | 0 | 0.0% |
| 300,000 to 400,000 | 3 | 11.5% |
| 400,000 to 500,000 | 6 | 23.1% |
| 500,000 to 600,000 | 2 | 7.7% |
| 600,000 to max | 2 | 7.7% |
| no zero window | 7 | 27.0% |
| Total | 26 | 100.0% |

the traffic pattern started appearing. There should have been a similar test performed for JAP, but only one cascade seemed usable for our research.

Table III shows the results of tests performed using a 630 KB file against different paths through Tor. Every measurement was performed after a Tor restart, and resulted in a different path being used. We show the number of bytes transferred from the connection start to the first packet with zero window size (precisely: a window size lower than 1400 bytes).

We found that there are big differences between the paths. The situation of the window size dropping to zero may appear after less than 100,000 bytes have been transferred, but may not appear in the whole transfer. We looked into the traces in more detail and found a number of interesting facts. There were differences in the initial window. Also, the window size reached close to the beginning of the connection differs between the traces. A large window size appeared in the transmissions with a non-zero window, or in those with a zero window close to the transfer end.

That leads us to the conclusion that the differences are caused by the conditions on the path, including the configuration and operating system of the anonymity system nodes (especially the one closest to the server). It seems that the 'bursty' traffic is not caused by the buffers' overruns on the whole path, but rather as an effect of Tor's internal flow control.

It is also worth noting that if the zero window is reached, the bursts appear regularly for the rest of the transfer.

As the initial window size differs between the nodes, so does the amount of data that may be in transit simultaneously. That may lead to differences in response time, for applications that have such needs.

*Artifacts:* We observed a number of artifacts in the traffic. Examples include: oscillation at 220 sec in Figure 1a, disturbance between 350 and 450 sec in Figure 1 with its' echo in Figure 1, bursts on higher bandwidth transmissions in Figure 4 or oscillations in Figure 9.

While searching for possible explanations, we analysed the source code of the two tested systems. There was no direct answer. However, we have found out that, as data from many flows travel by a single TCP connection between the internal nodes, an retransmission or any other problem will affect performance on multiple streams at the same time. It means that the disturbances may be an effect of interactions with other flows passing through the anonymity systems.

## V. Conclusions

We have shown a number of network-level effects that affect performance of currently deployed, TCP-based anonymity systems. We have also presented results showing that the effects may be visible in practice.

Our research shows that describing the network properties of an anonymity system by using only delay is not enough, and there are more factors to consider. We have also shown how the internal design affects flow characteristics. The results can be interpreted in two ways. The first shows possible performance issues, and the second one possible attacks (as the traffic may be marked from both sides of the connection).

We have shown that both systems do not change the traffic characteristics to the degree that would allow them to hide fast rate changes. Bursts of a length of 60 seconds were recognisable after passing by both systems. Additionally, that was also true for JAP and 10 second bursts. This limits the space for attacks that would watermark the traffic by bandwidth changes, like in [21]. Also, both systems tend to output longer bursts than the input bursts.

It should be noted that receiver behaviour (instead of only sender behaviour) should also be taken into account when modelling anonymity systems using network protocols with flow control like the TCP. When the end-to-end property is broken, characteristic flow bursts may appear on the sender side. How it may be used by an attacker remains a problem for future work to examine. It clearly affects performance, however, and introduces delay when a fast reply is needed. Adding to, or improving application level flow control in anonymity systems, could be a solution to this problem. Such mechanism should be designed to find a compromise between performance and the time of reaction to events from the other peers.

That may also mean that different settings or implementation differences of the network protocols could have a noticeable impact on the anonymity system's performance.

## References

[1] Alleman, M., Paxson V., and Stevens, W, *TCP Congestion Control*, RFC 2581, April 1999.

[2] Berthold, O. Federrath, H., Köpsell, S, *Web MIXes: A system for anonymous and unobservable Internet access*, Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, 115–129, July 2000, Springer-Verlag, LNCS 2009.

[3] Boucher, P., Shostack, A., Goldberg, I., *Freedom Systems 2.0 Architecture*, white paper, Zero Knowledge Systems, Inc., December 2000.

[4] Clark, D., *Window Acknowledgement Strategy in TCP*, RFC 813, July 1982.

[5] Danezis, G. *Statistical Disclosure Attacks: Traffic Confirmation in Open Environments*, Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003), pp. 421–426, May 2003.

[6] Danezis, G., Dingledine, R., Mathewson, N., *Mixminion: Design of a Type III Anonymous Remailer Protocol*, Proceedings of the 2003 IEEE Symposium on Security and Privacy, May 2003.

[7] Dingledine, R., Mathewson, N., Syverson, P. *Tor: The Second-Generation Onion Router*, Proceedings of the 13th USENIX Security Symposium, August 2004.

[8] I2P site, URL: http://www.i2p.net.

[9] Jacobson, V., Braden, B., Borman, D., "TCP Extensions for High Performance", RFC 1323, May 1992.

[10] "Technischer Hintergrund von JAP", technical paper, URL: http://anon.inf.tu-dresden.de/develop/JAPTechBgPaper.pdf

[11] Köpsell, S., "AnonDienst—Design und Implementierung", technical paper, http://anon.inf.tu-dresden.de/develop/Dokument.pdf

[12] Levine, B. N., Reiter, M. K., Wang, C., Wright, M. K., *Timing Attacks in Low-Latency Mix-Based Systems*, Proceedings of Financial Cryptography (FC '04), February 2004, LNCS 3110, Springer-Verlag.

[13] Möller, U., Cottrell, L., Palfrader, P., Sassaman, L. "Mixmaster Protocol—Version 2", July 2003.

[14] Murdoch, S. J., Danezis, G., *Low-Cost Traffic Analysis of Tor*, Proceedings of the 2005 IEEE Symposium on Security and Privacy, May 2005, IEEE CS.

[15] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[16] Serjantov, A., Sewell, P. *Passive Attack Analysis for Connection-Based Anonymity Systems*, Proceedings of ESORICS 2003, October 2003.

[17] Shmatikov, V., Wang, M.-H., *Timing Analysis in Low-Latency Mix Networks: Attacks and Defences*, Proceedings of ESORICS 2006, September 2006.

[18] Wendolsky, R., Herrmann, D., Federrath, H., *Performance Comparision of low-latency Anonymisation Services from User Perspective*, Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007), June 2007, Springer.

[19] Wiangsripanawan, R., Susilo, W., Safavi-Naini, R., *Design principles for low latency anonymous network systems secure against timing attacks*, Proceedings of the fifth Australasian symposium on ACSW frontiers (ACSW '07), 2007.

[20] Tóth, G., Hornák, Z., *Measuring Anonymity in a Non-adaptive, Real-time System*, Proceedings of Privacy Enhancing Technologies workshop (PET 2004), LNCS 3424, Springer-Verlag.

[21] Yu, W., Fu, X., Graham, S., Xuan, D., Zhao, W., *DSSS-Based Flow Marking Technique for Invisible Traceback*, SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp. 18–32, 2007, IEEE Computer Society.

[22] Zhu, Y., Fu, X., Graham, B., Bettati, R., Zhao, W., *On Flow Correlation Attacks and Countermeasures in Mix Networks*, Proceedings of Privacy Enhancing Technologies workshop (PET 2004), LNCS 3424.

[23] Zhu, Y., Bettati, R., *Unmixing Mix Traffic*, Proceedings of Privacy Enhancing Technologies workshop (PET 2005), May 2005.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Performance Evaluation of a Machine Learning Algorithm for Early Application Identification

Giacomo Verticale and Paolo Giacomazzi
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Italy
{vertical,giacomaz}@elet.polimi.it

***Abstract*—The early identification of applications through the observation and fast analysis of the associated packet flows is a critical building block of intrusion detection and policy enforcement systems. The simple techniques currently used in practice, such as looking at the transport port numbers or at the application payload, are increasingly less effective for new applications using random port numbers and/or encryption. Therefore, there is increasing interest in machine learning techniques capable of identifying applications by examining features of the associated traffic process such as packet lengths and interarrival times. However, these techniques require that the classification algorithm is trained with examples of the traffic generated by the applications to be identified, possibly on the link where the the classifier will operate. In this paper we provide two new contributions. First, we apply the C4.5 decision tree algorithm to the problem of early application identification (i.e. looking at the first packets of the flow) and show that it has better performance than the algorithms proposed in the literature. Moreover, we evaluate the performance of the classifier when training is performed on a link different from the link where the classifier operates. This is an important issue, as a pre-trained portable classifier would greatly facilitate the deployment and management of the classification infrastructure.**

## I. Introduction

THERE is a constant growth of new Internet applications using either random transport port numbers or reusing well-known ports registered to other applications. For example, peer-to-peer applications do not require the usage of well-known ports and some peer-to-peer applications, such as Skype, use port hopping. Moreover, recent applications frequently tunnel traffic through HTTP connections to seamlessly cross firewalls and NAT boxes. In these cases, the application generating a traffic flow cannot be identified by simply looking at the transport ports.

Therefore, there is a growing interest for alternative classification algorithms relying on features different from transport ports. The most widespread approach relies on the inspection of the packet payload and on the matching with signatures characteristics of the applications to be identified. This solution has the drawback of requiring computationally heavy elaborations that must be made at wire-speed; further it is not effective on cyphered traffic, such as that generated by applications secured through SSL.

Other approaches use the information available in the non-cyphered IP packet header and complement these data with the analysis of statistical properties of the packet flow, considered as a random process. An example of this type of additional metrics is the packet interarrival time. With this approach, each traffic flow is associated with a set of features and, by examining the measured values of these features, a classifier yields the most likely application associated to the flow.

In [1] and [2] the authors use a Bayesian method to classify a traffic flow by using metrics such as flow duration, flow bandwidth, and statistics on packet sizes and interarrival times. The main drawback of this approach is that it is possible to classify only terminated flows. In [3] and [4], the authors introduce the idea of *Early Application Identification* and classify traffic flows by looking only at the lengths of the first packets in a flow; classification is performed by using clustering techniques such as K-means, Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM). The authors of [5] further extend the idea by including packet interarrival times in the feature set and by developing a new classification algorithm based on the Bayesian method. Finally, a preliminary comparison of general-purpose algorithms is given in [6], and the C4.5 algorithm [7] is indicated as one of the best performing. However, in [6] the authors study the classification of terminated flows and early application identification is not considered.

Using Machine Learning techniques for application identification still presents some challenges. In order to train the classifier it is necessary to collect a representative set of flow instances whose applications are known in advance. For example, by using offline payload inspection techniques to label the training data. The main issue with this approach is that the statistical properties of the traffic flows vary from link to link, therefore we expect that, if we train a classifier at a given link and operate it at another link, performance will be worse.

Another problem is that the flow must be observed for some time, in order to collect data enough to categorize it. In the case of early identification a trade-off must be found between speed and accuracy.

This paper presents several contributions along the line of assessing the capability of state-of-the-art Machine Learning techniques for fast traffic classification. The key contributions of the paper are the following.

Work on this paper has been supported by the Italian Ministry of Research project RECIPE

First, we propose the C4.5 algorithm in the context of early application identification and we compare its accuracy to the results reported in [2], [4], and [5] and obtain superior results in terms of both true and false positives.

Second, we observe that the classification performance obtained by observing as few as 5 packets per direction is very similar to the performance obtained by observing all the packets in a flow, provided that suitable features, such as the actual packet lengths, are chosen for classification.

Third, we evaluate the performance of a classifier trained in a WAN link different than the one where it is operated. The performance is worse than using the classifier in the same link, but not to a large extent. Further, there are significant performance differences among protocols; in particular, the classification of HTTP protocol is very robust, whereas the Telnet and FTP traffic flows are more difficult to identify.

The paper is organized as follows. Section II explores the state-of-the-art of traffic classification using Machine Learning techniques. Section III discusses the data and the tools we use in the paper and explains the experimental setup. Section IV reports and discusses the results of our investigation. Finally, the last Section yields our concluding remarks.

## II. Basic Concepts and Related Work

We identify two main application areas of Internet traffic classification: the separation of malicious traffic from good traffic and the identification of the application that generates the packets. Today, deep packet inspection boxes are used for both objectives. For example SNORT [8] for identification of malicious traffic and l7filter [9] for application identification. Both packages perform per-flow classification by comparing the payload of the first packets in the flow with a set of pre-configured signatures, which are generally human-made.

Deep packet inspection is not adequate when traffic is cyphered or when high-speed links are considered, therefore there is a growing research interest for traffic classification considering only packet lengths, interarrival times and any other information available in the protocol headers. One of the first proposals of application identification through the examination of the traffic process is made in [10], where the authors point out that RealAudio traffic exhibits a behavior significantly different from that of telnet, HTTP or FTP traffic, in particular in terms of flow duration and regularity of packet lengths and interarrival times. The authors, however, do not propose a specific classification algorithm.

A more recent work proposes to use clustering techniques for mapping a traffic flow to a QoS class [11]. The authors find that the most significant metrics are the average packet size and the flow duration. The main problem with these metrics is that classification can be performed only when the flow is already finished, limiting the practical utility of this technique.

The authors of [12] apply unsupervised Bayesian techniques to the problem of application identification and use a feature selection technique to to find the optimal set of attributes. The authors find that the most influential attributes are the mean and the variance of the packet lengths, the total amount of traffic in the flow, and the flow duration. Also the mean interarrival time has some influence. As for [11], not all these metrics are suitable for early application identification.

In [1], the authors propose a supervised Bayesian classifier and also use some TCP-specific metrics such as the count of all the packets with the PUSH bit set. However, only semantically complete TCP connections are considered.

Paper [13] proposes how to use behavior analysis to augment the efficacy of automatic classification techniques. Their BLINC framework considers attributes such as the social and functional role of the hosts, which require long observation times to elaborate and, therefore, making BLINC a valuable solution for off-line classification.

In [6], the authors compare the performance of five general purpose supervised Machine Learning algorithms, showing that the C4.5 tree based algorithm provides a good trade-off between classification accuracy and speed. The authors also consider two feature reduction schemes and identify two feature sets that show similar performance. In one feature set packet length statistics are predominant, while in the second feature set there is a balance of packet lengths and interarrival times. The authors of [6] have also developed a set of tools [14] which automates the process of collecting packet traffic, reconstructing the flows, computing metrics and invoking the Machine Learning algorithms.

The authors of [3] study the problem of Early Application Identification and show that it is enough to know the length of the first four or five packets to achieve promising classification results. The authors also compare three supervised clustering techniques.

Finally, the authors of [5] further refine the idea, by considering also packet interarrival times and developing a novel classification algorithm, but do not compare it to other state-of-the-art classification algorithms.

## III. Experimental Setup

In our research work we have set up a laboratory to experiment in practice the classification algorithms. We have used publicly available software and data. Packet traces representative of WAN traffic have been retrieved from the NLANR traffic archive [15]. In particular we have used the `auckland-vi-20010611`, `auckland-vi-20010612`, and `nzix-ii-20000706`. The two `auckland` traces were collected at the same network link; in the paper, we will use the first trace to train the Machine Learning classifier, whereas the second trace will be used to assess the classification performance. We will refer to the first trace as *Samplepoint A (training)* and to the second trace as *Samplepoint A (test)*. The third trace was collected at a different link, and it will be referred to as *Samplepoint B (test)*.

In order to group packets in flows and to elaborate per-flow metrics, we have used the `NetMate`[16] software, extended with the `NetAI`[14] patch. This way, we obtain the features reported in Table I, which we will refer to as the two *Standard* sets. We have isolated the *Standard 2* features because they

TABLE I
THE *Standard* FEATURES

| **Standard 1 features** |
|---|
| Min, max, mean and std. deviation of the packet lengths in the forward direction. |
| Min, max, mean and std. deviation of the packet lengths in the backward direction. |
| Min, max, mean and std. deviation of the packet interarrival times in the forward direction. |
| Min, max, mean and std. deviation of the packet interarrival times in the backward direction. |
| Transport protocol number. |
| Total number of TCP URG and PUSH flags in the forward direction. |
| Total number of TCP URG and PUSH flags in the backward direction. |
| **Standard 2 features** |
| Flow duration. |
| Total number of bytes and of packets in the forward direction. |
| Total number of bytes and of packets in the backward direction. |

TABLE II
THE *Extended* FEATURES

| **Extended 1 features** |
|---|
| Lengths of the first 3 packets in the forward direction |
| Lengths of the first 3 packets in the backward direction |
| Interarrival times of the first 3 packets in the forward direction |
| Interarrival times of the first 3 packets in the backward direction |
| **Extended 2 features** |
| Lengths of the first 5 packets in the forward direction |
| Lengths of the first 5 packets in the backward direction |
| Interarrival times of the first 5 packets in the forward direction |
| Interarrival times of the first 5 packets in the backward direction |

are not meaningful when only the first packets of the flow are considered.

We have also implemented a new patch for NetMate to collect the feature sets proposed in [3] and [5], referred to as the *Extended* feature sets (Table II). These extended sets are the actual packet lengths and the interarrival times of the first packets of the flows. We have grouped the features in two sets of increasing size. The *Extended 1* feature set includes the packet lengths of the first three packets of a flow in each direction and the two interarrival times between them in each direction, for a total of 10 features. The *Extended 2* feature set includes all the packet lengths and interarrival times involving the first 5 packets in a flow.

As a concluding remark, we note that the NetMate software defines a flow as the packets belonging to a single TCP connection, in case TCP is the transport protocol. In case UDP is used, the flow is defined as all the packets with the same IP addresses and UDP port numbers and considered finished when no packets have arrived for 600 s. From all the flows available in the data sets, we have randomly chosen 4000 flows for each of the five application protocols considered in this paper. For some protocols there were fewer flows than required, so we used all the available ones. The total number of flows chosen in each data set is reported in Table III.

TABLE III
SIZE OF THE DATA SETS

| Data Set | Number of Flows | Flows with at least 5 packets per direction |
|---|---|---|
| Samplepoint A (training) | 15606 | 15300 |
| Samplepoint A (test) | 15803 | 15545 |
| Samplepoint B (test) | 13335 | 12788 |

Then, we have processed the selected flows and flow features by using the Weka [17] machine learning suite to train the classifier and perform the validation tests. We have preliminarly evaluated the same algorithms as in [6] and concluded that the C4.5 algorithm [7] gives the best results, so we will show only the results obtained with that algorithm. In fact, the C4.5 algorithm well suits problems in which several attributes assume discrete values and when training data is noisy, which is common in large data sets. To ease a comparison, we performed our assessment by using the same 5 applications as in [6], i.e. FTP-data, Telnet, SMTP, DNS, and HTTP. For the training and for the validation we assumed that the flow category label is the server well-known port number.

## IV. EXPERIMENTAL RESULTS

In this section, we assess the performance of Machine Learning techniques for traffic classification and substantiate the following findings:

- the C4.5 algorithm is superior to the algorithms proposed in [2], [4], and [5];
- the classification accuracy obtained with five packets per direction is similar to the accuracy obtained observing all the packets in the flow even if the trained classifier is used on a different link, but only if the Extended Features are used;
- the performance loss measured when the trained classifier is used on a different link is mainly due to a few protocols, whereas other protocols, in particular HTTP, show minimal performance differences.

### A. C4.5 Algorithm for Early Application Identification

As a first result, we show that the the state-of-the art C4.5 algorithm [7] improves the classification performance over the algorithm proposed in the literature. As performance metrics we use the True Positive Rate and the False Positive Rate, defined as:

$$\mathrm{TPR}(i) = \frac{e_i}{E_i}$$
$$\mathrm{FPR}(i) = \frac{\bar{e}_i}{e_i + \bar{e}_i}$$

where $E_i$ is the number of instances of protocol $i$ in the evaluation set, $e_i$ is the number of instances of protocol $i$ correctly classified as $i$, and $\overline{e_i}$ is the number of instances of other protocols incorrectly classified as $i$.

In Table IV, we compare the True Positive Rates as reported in [2], [4], [5], as well as the test results obtained using the C4.5 classifier trained on the *Samplepoint A (training)* data and tested on the *Samplepoint A (test)* data. In our experiment we used the *Standard-1* plus *Extended-2* feature sets. In Table V, we make a similar comparison considering the False Positive Rates. In reporting results from other papers, we have considered only the classes considered in at least two papers.

TABLE IV
COMPARISON OF THE TRUE POSITIVE RATES

| Protocol | [2] | [4] | [5] | Our solution |
|---|---|---|---|---|
| HTTP | 89.2% | 96.2% | 91.8% | 99.7% |
| SMTP | *97.2% | 90.1% | 94.5% | 98.6% |
| POP3 | *97.2% | 93.4% | 94.6% | N/A |
| FTP | 97.9% | 92.4% | N/A | 94.8% |

*In [2] STMP and POP3 form a single class

TABLE V
COMPARISON OF THE FALSE POSITIVE RATES

| Protocol | [2] | [4] | [5] | Our solution |
|---|---|---|---|---|
| HTTP | 10.4% | 1.3% | 6.4% | 0.1% |
| SMTP | *2.3% | 0.1% | 3.1% | 1.4% |
| POP3 | *2.3% | 0.7% | 3.1% | N/A |
| FTP | 1.8% | 0.4% | N/A | 0.5% |

*In [2] STMP and POP3 form a single class

Our solution shows higher true positive rates than the other solutions for HTTP and SMTP and is inferior to [2] only for the FTP class, where, however, our solution shows a much lower FPR. Our proposal shows false positive rates superior to [2] and [5]. Compared to [4], our solution performs similarly for the FTP class and worse only for the SMTP class, where our solution scores a much higher TPR.

Finding a good trade-off between TPR and FPR is a common challenge when using Machine Learning techniques. In the context of internet traffic classifications for monitoring and security purposes, it is important to achieve a low FPR. From Tables IV and V, we conclude that the C4.5 classifier shows an FPR performance comparable to the best other solution ([4]), but with a higher TPR. Therefore, in the rest of the paper we will consider only the C4.5 algorithm.

### B. Effect of the Observation Horizon and of the Feature Sets

Table VI reports the percentage of incorrectly classified traffic flows for different observation horizons and different feature sets. Results are given for three data sets. The column labeled *Samplepoint A(training)* contains the percentage of flows in the training set equivocated by the classifier, which gives us a lower bound on the classification error that we expect at run time. In all the cases, when we consider at least the first 5 packets per direction, the residual error is lower than 0.3%, and is about 0.5% if we consider only the first 3 packets per direction.

In Table VI the column labeled *Samplepoint A (test)* gives the observed percentage of incorrectly classified packets when the trained classifier is used on a different data set collected on the same WAN network link. Again, when we consider the first 5 or more packets per direction, the error does not change and settles at about 1.5%. On the other hand, if we consider only the first 3 packets, the error is more than 3.5%. This trend is in line with [3] and [5], which indicate in 4 or 5 packets the ideal observation horizon.

The last column gives the classification error when the trained classifier is used with a data set collected on a completely different WAN network link. There are no data in the literature about this problem, but we expect that the different statistics of the collected features hamper the work of the classifier. Results show that the error grows more than 3.5 times, if we consider only the standard set and 3, 5, or 10 packets. Slightly better results are obtained if all the packets of the flow are considered, with an error about 4.5%, which is only 3 times larger than in the *Samplepoint A(test)* case. On the other hand, the Extended Set greatly improves the robustness of the classifier, bringing the error from about 6% down to about 4%, which is about 2.5 times the result obtained in the *Samplepoint A(test)* case and is better than the result obtained with the standard set and observing all the packets in the flow. Further, these apparently non promising results should not make us abandon the idea of using this kind of classifiers because, as we show later, the vast majority of the incorrectly classified packets come from a limited amount of protocols.

TABLE VI
INCORRECTLY CLASSIFIED INSTANCES USING THE *Standard* AND *Extended* FEATURES

| | | Samplepoint | | |
|---|---|---|---|---|
| Packets per direction | Feature set | A (training) | A (test) | B (test) |
| 3 | std1 | 0.48% | 3.52% | 13.54% |
| 3 | std1 + ext1 | 0.44% | 3.22% | 13.06% |
| 5 | std1 | 0.27% | 1.55% | 6.01% |
| 5 | std1 + ext2 | 0.22% | 1.80% | 4.10% |
| 10 | std1 | 0.21% | 1.66% | 6.01% |
| all | std1 + std2 | 0.21% | 1.45% | 4.45% |

TABLE VII
ACCURACY BY CLASS WHEN USING THE *Standard 1* AND *Extended 2* FEATURES

| | Samplepoint A | | Samplepoint B | |
|---|---|---|---|---|
| Class | TPR | FPR | TPR | FPR |
| DNS | 100% | 0.4% | 99% | 0.1% |
| FTP (data) | 95% | 0.5% | 77% | 1.4% |
| Telnet | 92% | 0.1% | 84% | 0.4% |
| SMTP | 98% | 1.4% | 95% | 3.0% |
| HTTP | 99% | 0.1% | 99% | 0.2% |

### C. Per-protocol Classification Accuracy

Finally, Table VII shows the classification performance obtained on test data, broke down by protocol. All the results are obtained considering 5 packets per direction and the *Standard* plus the *Extended 2* metrics. As in the experiment above, the classifier was trained with the *Samplepoint A (training)* dataset and operated with the *Samplepoint A (test)* and *B (test)* datasets.

As already observed in [3] and [5], there are significant performance differences among protocols. Our results, however, shed light on what protocols fingerprints are most likely to be preserved from link to link.

In *Samplepoint A*, the true positives are always reasonably good, with only FTP and Telnet equal to or below 95%. Porting the classifier to *Samplepoint B* worsens the performance of these two protocols, whose TPR drops below 85%, while the other protocols maintain their good performance. Regarding the false positives, the only class with FPR larger than 1% is

SMTP in the *Samplepoint A*, whereas, in the *Samplepoint B* case, both SMTP and FTP have a large FPR.

At the light of these results, we remark that, when porting a trained classifier, those protocols that had a lower TPR tend to decrease their TPR and those protocols that had a higher FPR tend to increase it. However, there can be anomalies, for example in our experiment DNS and FTP had similar FPR in the first link and very different FPR in the second.

## V. Conclusions

In our research work we have experimented, with a thorough laboratory activity, the behavior of machine learning algorithms for the classification of applications by examining the traffic flow process. We have concentrated our attention to the early application identification, therefore, we have examined extended feature sets including lengths and interarrival times of the first few packets of flows and we have matched their performance against that of standard feature sets requiring the examination of the entire traffic flow.

We have examined the behavior of the C4.5 decision tree algorithm with extended feature sets and we have determined, as a novel result, that this algorithm performs very well for early application identification.

We have proceeded by studying a problem so far left open in the related research, that is, the portability of a trained Machine Learning classifier on a link different from that used for training. The possibility of porting pre-trained classifier would be very appealing for practical implementations. We have determined that the performance of a trained classifier moved to another link worsens, but the degradation seems to be concentrated on specific protocols such as FTP and Telnet, while other protocols such as HTTP and DNS are recognized effectively even by a moved classifier.

Our current work concentrates on devising feature sets capable of improving the portability of trained classifiers to different links.

## References

[1] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2005, pp. 50–60.

[2] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for internet traffic classification," *Neural Networks, IEEE Transactions on*, vol. 18, no. 1, pp. 223–239, Jan. 2007.

[3] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, 2006.

[4] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *The 2nd ADETTI/ISCTE CoNEXT Conference*, Dec. 2006.

[5] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 5–16, 2007.

[6] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, 2006.

[7] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[8] "SNORT," http://www.snort.org/.

[9] "L7filter: Application Layer Packet Classifier for Linux," http://l7-filter.sourceforge.net/.

[10] J. Mena, A.; Heidemann, "An empirical study of real audio traffic," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 101–110 vol.1, 2000.

[11] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2004, pp. 135–148.

[12] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," *lcn*, vol. 0, pp. 250–257, 2005.

[13] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 229–240, 2005.

[14] "netAI, Network Traffic based Application Identification," http://caia.swin.edu.au/urp/dstc/netai/.

[15] "NLANR traces," http://pma.nlanr.net/Special/.

[16] "NetMate Meter," http://sourceforge.net/projects/netmate-meter/.

[17] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementations*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000.

# Workshop on Wireless and Unstructured Networking: Taming the Stray

AS NETWORK customers, operators, and designers strive to bring more productivity and enjoyment into increasingly diverse areas of our lives, be it business and commerce, health services, crisis management, content distribution, or multiuser games, mobile communications and ubiquitous computing are becoming the leading interactivity paradigm. While being thus “condemned” to succeed, they raise at least two main challenges. One is the multitude of standards specifying reliable high-speed wireless networking solutions, and the resultant multitude of heterogeneous technologies that follow (IEEE 802.11, 802.15 and 802.16, Bluetooth, MANET, SANet, Mobile IP, and others). Making them coexist, let alone cooperate, requires integration of many diverse areas of research, such as internetworking, distributed computing, signal processing, networking theory, and economics. Can we venture to predict the outcome? Can we make a case for particular scenarios? Another challenge is the dualism of fixed-infrastructure vs. ad hoc networks. The latter, despite their 20-year presence in academic literature, have been remarkably slow to materialize in the real world. Attempts to blame this on our natural distrust towards systems under distributed control and lacking clear ownership do not sound convincing: P2P technology does away with central administration, yet it has become successful to the point of embarrassing network operators and intellectual property managers. Can ad hoc networks repeat the success of P2P without becoming an embarrassment? The purpose of this workshop is to address questions like the above, to look at wireless networking from a broader perspective, to present ideas and share experience from their verification.

Here is the non-exclusive list of topics:

- routing and location protocols
- MAC schemes
- modeling methodologies for wireless channels
- cross-layer issues
- anonymity and security
- performance studies and comparisons
- mobile applications
- mobility support, handover mechanisms, mobility brokers (MB), MIP
- self-organization
- residential access systems,
- interworking between ad-hoc and fixed/mobile networks
- virtual networks
- ad-hoc networks as part of urban mesh,
- content distribution and multicast overlay
- pricing and incentive arbitration
- software supporting networked applications
- case studies

International Programme Committee

**IYuh-Shyan Chen,** National Taipei University, Taiwan

**Franco Davoli,** University of Genoa, Italy

**Janelle Harms,** University of Alberta, Canada

**Jerzy Konorski,** Gdansk University of Technology, Poland

**Pawel Matusz,** Anritsu, UK

**Marek Natkaniec,** AGH—University of Science and Technology, Poland

**Ioanis Nikolaidis,** University of Alberta, Canada

**Wladek Olesinski,** Sun Microsystems, USA

**Wlodek Olesinski,** Olsonet Communications Corporation, Canada

**Piotr Pacyna,** AGH—University of Science and Technology, Poland

**Andrzej Pach,** AGH—University of Science and Technology, Poland

**Joon-Sang Park,** Hongik University, Korea

**Krzysztof Szczypiorski,** Warsaw University of Technology, Poland

**Slawomir Kuklinski,** Warsaw University of Technology, Polandrope, Germany

Workshop Chairs

**Krzysztof Szczypiorski,** Warsaw University of Technology, Poland

**Konrad Wrona,** NATO C3 Agency, The Netherlands

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Supporting Wireless Application Development via Virtual Execution

Nicholas M. Boers, Pawel Gburzyński, Ioanis Nikolaidis
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
{boers,pawel,yannis}@cs.ualberta.ca

Wlodek Olesinski
Olsonet Communications
51 Wycliffe Street
Ottawa, Ontario, Canada K2G 5L9
wlodek@olsonet.com

***Abstract*—We introduce our "holistic" platform for building wireless ad hoc sensor networks and focus on its most representative and essential virtualization component: VUE$^2$ (the Virtual Underlay Emulation Engine). Its role is to provide a vehicle for authoritative emulation of complete networked applications before physically deploying the wireless nodes. The goal is to be able to verify those applications *exhaustively* before programming the hardware, such that no further (field) tests are necessary. We explain how VUE$^2$ achieves this goal owing to several facilitating factors, most notably the powerful programming paradigm adopted in our platform. As implied by the holistic nature of the discussed system, our presentation touches upon operating systems, simulation, network protocols, real-time systems, and programming methodology.**

## I. Introduction

ALTHOUGH simple wireless devices built from low-end components are quite popular these days, one seldom hears about serious wireless networks within this framework. While it is not a big challenge to implement simple broadcasters of short packets, it is quite another issue to turn them into collaborating nodes of a serious ad hoc wireless system. Apparently, many popular ad hoc routing schemes proposed and analyzed in the literature [1]–[6] address devices with a somewhat larger resource base. To make matters worse, some people believe that such devices must be programmed in Java to make serious applications possible [7].

In this context, efforts to introduce an order and methodology into programming small devices often meet with skepticism and shrugs, the common misconception being that one will not have to wait for long before those devices disappear and become superseded by larger ones capable of supporting "serious" programming platforms. This is not true. Despite the ever decreasing cost of microcontrollers, we see absolutely no reduction in the demand for the ones at the lowest end of the spectrum. On the contrary: their low cost and power requirements enable new applications and narrow the gap between the publicized promise of ad hoc wireless sensor networking and the actual state of affairs. Similar to the impossibility of having the three desirable properties of food (cheap, fast, and good tasting) present at the same time, wireless sensor networking has problems being cheap, ad hoc, and useful, all at once.

The goal of our platform is to enable the rapid creation of wireless ad hoc sensor networks using the smallest and cheapest devices available today. In our networks, such devices are capable of ad hoc routing while offering enough processing power to cater to complex applications involving distributed sensing and monitoring. To thoroughly test and evaluate our work, we turn to virtualization.

At the lowest level, virtualization can be accomplished by simulating in software a particular instruction set, i.e., by means of a bytecode interpreter. This approach has been the basis even for commercial grade products, but within the scope of our paper the primary example is Maté [8]. The main shortcomings of such an approach are (a) the interpreter overhead (in terms of both space and time), (b) a lack of back ends for compilation from familiar high-level languages into the invented bytecode format, and (c) the need to define the interaction with peripheral components, e.g., transceivers, in a manner consistent with the invented instruction set. If the virtual machine is fairly well established, e.g., JVM, then one can claim that at least (b) and (c) have been addressed. However, there seems to exist no successful virtual machine geared to sensor devices. For example, even virtual machines intended for small footprint devices (like Dalvik VM [9], part of the Google Android platform [10]) are "heavyweight" for sensor nodes. In addition (as it happens with Dalvik VM), not all of the language libraries are implemented, raising questions on the extent that VMs can be ported to small platforms without compromising fidelity.

On the other end of the spectrum, we find virtualization by means of an API provided by the underlying operating system. The API is accessible using familiar high-level languages like C or C++. One example is the POSIX API. A platform that can provide run-time emulation of an API can be thought of as successfully virtualizing at the level of the API. Unfortunately, commodity OS APIs are very broad, and the abstractions that they promote are expensive to implement on a sensor device. For example, the Berkeley sockets API is a powerful, albeit expensive, way to abstract network communication. Even worse, it is not a useful abstraction for small devices that do not even implement a TCP/IP stack.

We believe that a viable compromise between the two extremes is to introduce a small footprint OS, to specify the API supported by the OS, and to subsequently offer virtualization at the level of that API. Note that in following this approach, we do not need a new toolchain for code production, since we neither have to invent a new higher level language nor do we need to procure a compiler back end for a new (invented) instruction set. In this paper, we focus on the interplay of PicOS (our operating system for tiny microcontrolled devices [11]) and VUE$^2$ (the Virtual Underlay Emulation Engine for realistically simulating networked applications programmed in PicOS).

## II. PicOS

The most serious problem with implementing non-trivial, structured, multitasking software on microcontrollers with limited RAM is minimizing the amount of memory resources needed to sustain a thread. The most troublesome component of the thread footprint is its stack, which must be preallocated to every thread in a safe amount sufficient for its maximum possible need.

PicOS strikes a compromise between the complete lack of threads and overtaxing the tiny amount of RAM with fragmented stack space. A thread contains a number of *checkpoints* that provide preemption opportunities. In the imposed structured organization of a thread, we try to (a) avoid locking the CPU at a single thread for an extensive amount of time and (b) use the checkpoints as a natural and useful element of a thread's specification to enhance its clarity and reduce its structure's complexity. These ideas lie at the heart of PicOS's concept of threads, which are structured like finite state machines (FSMs) and exhibit the dynamics of coroutines [12], [13] with multiple entry points and implicit control transfer.

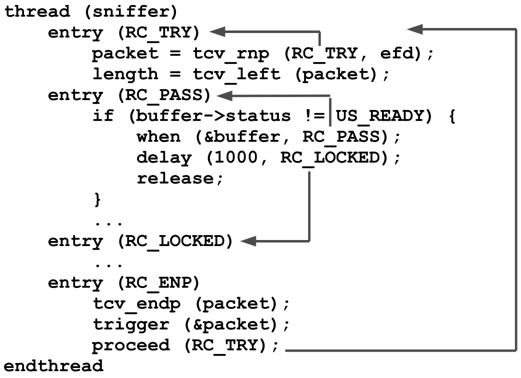
The value of FSM-like programming abstractions is evident to anyone developing networking protocols, as most protocols tend to be described, or even formalized, as communicating FSMs. In addition, programming using a coroutine paradigm is a fairly well-accepted approach and has survived in modern languages (e.g., "stackless" Python [14] and more recently Ruby [15]). The only general criticism coroutines receive is that the lack of arbitrary preemption might allow CPU-bound tasks to monopolize the CPU. This is not a fundamental problem, because a CPU-bound task can be broken down into a sequence of states to allow preemption at state transitions. However, an important element of our view is to use the coroutine paradigm as a means to *discourage* CPU-intensive tasks on sensors. Instead, any tasks of this kind should be either moved to data collectors (i.e., full-scale computers) or delegated to specialized (possibly reconfigurable) hardware.

On a historical note, we arrived at PicOS indirectly as a step in the evolution of our network simulation system SMURPH [16], [17]. As SMURPH underwent a number of enhancements, its capability for rigorous representation of all the relevant engineering problems occurring in detailed protocol design turned it into a specification system. Although oriented towards modeling networks and their protocols, SMURPH became a *de facto* general purpose specification and simulation package for reactive systems [18], [19].

A highly practical project—the development of a low-cost wireless badge—inspired the idea to implement a complete executable environment for microcontrollers based on SMURPH's programming paradigm. After developing a SMURPH model for the badge, the most reliable way of transporting it to the real device was to implement the target microprogram on top of a tiny execution environment mimicking SMURPH's mechanism for multithreading and event handling [11]. Incidentally, that mechanism facilitated a stackless implementation of multithreading. Consequently, the resultant footprint of the complete application was trivially small ($<$ 1 KB of RAM), while the application itself was expressed as a structured and self-documenting program strictly conforming to its SMURPH model.

### A. The anatomy of a PicOS thread

Fig. 1 shows a sample PicOS thread. In this C code, new keywords and constructs are straightforward macros handled by the standard C preprocessor. The `entry` statements mark the different states of the thread's FSM.



```
thread (sniffer)
    entry (RC_TRY)
        packet = tcv_rnp (RC_TRY, efd);
        length = tcv_left (packet);
    entry (RC_PASS)
        if (buffer->status != US_READY) {
            when (&buffer, RC_PASS);
            delay (1000, RC_LOCKED);
            release;
        }
        ...
    entry (RC_LOCKED)
        ...
    entry (RC_ENP)
        tcv_endp (packet);
        trigger (&packet);
        proceed (RC_TRY);
endthread
```

Fig. 1: Code for a sample PicOS thread.

A thread can lose the CPU when (a) it explicitly relinquishes control at the boundary of its current state (e.g., `release`) or (b) a function call blocks (e.g., `tcv_rnp` if no new packets are available). In both cases, the CPU returns to the scheduler, which can then allocate it to another thread. Whenever a thread is assigned the CPU, execution continues in that thread's *current state*.

Before executing `release`, a thread typically issues a number of *wait requests* identifying one or more events to resume it in the future (e.g., `when` for IPC and `delay` for timed events). The collection of wait requests issued by a thread in every state describes the dynamic options for its transition function from that state.

### B. System organization

The organization of PicOS is shown in Fig. 2. VNETI (Versatile NETwork Interface) acts as a layerless networking module, whereby the equivalents of "protocol stacks" are implemented as plug-ins. The set of operations available to plug-ins involve queue manipulations, cloning packets, inserting special packets, and assigning to them the so-called *disposition codes* representing various processing stages. Any protocol can be implemented within this paradigm, with TARP (our Tiny Ad hoc Routing Protocol [20], [21]) being the most prominent example. The modus operandi of VNETI is that packets are *claimed* by the protocol plug-ins as well as the physical interface modules (PHY) at the relevant moments of their life in the module's buffer space. There is no explicit concept of processing hierarchy, e.g., enforcing traditional layers; thus, packets in VNETI are handled "holistically."

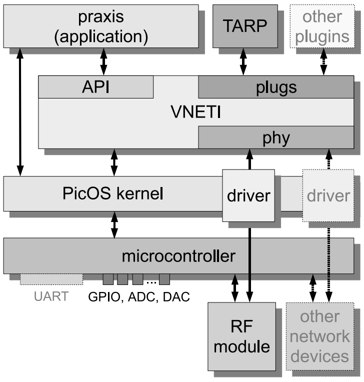


Fig. 2: The structure of PicOS.

All API functions interfacing the application (called the *praxis* in PicOS) to VNETI have the same status as those interfacing the praxis to the kernel, i.e., they are formally system calls. As a thread in PicOS can only be resumed at a state boundary, a potentially blocking system call requires a state argument (e.g., the first argument in the function call `tcv_rnp` in Fig. 1).

## III. VUE$^2$

The close relationship between PicOS and our discrete-time event-driven network simulator named SMURPH [16], [17] makes it possible to automatically transform PicOS praxes into SMURPH models with the intention of executing them virtually. VUE$^2$ implements the PicOS API within SMURPH, and in some cases, it can simply transform PicOS keywords into their SMURPH counterparts. To represent the physical environment of a PicOS praxis, it also provides a collection of event-driven interfaces. This way, a praxis can be compiled and executed in the environment shown in Fig. 3, with all the relevant physical elements of its node replaced by their detailed SMURPH models. Notably, exactly the same source code of VNETI is used in both cases.

### A. Time flow

The fidelity of the emulation environment depends to a great extent on appropriately handling the flow of time, i.e., equating emulated time with real time. In SMURPH, as in all event-driven simulators, the time tags associated with events are purely virtual. The actual (physical) execution time of a SMURPH thread is essentially irrelevant (unless it renders the model execution too long to wait for the results), and all that matters is the abstract delays separating the virtual events. For example, two threads in SMURPH may be semantically equivalent, even though one of them may exhibit a drastically shorter execution time than the other, e.g., due to more careful programming and/or optimization. In PicOS, however, the threads are not (just) models but they run the "real thing." Consequently, the execution time of a thread may directly influence the perceived behavior of the PicOS node. In this context, the following two assumptions made the VUE$^2$ project worthwhile:

1) PicOS programs are reactive, i.e., they are practically never CPU bound. The primary reason why a PicOS thread is making no progress is that it is waiting for a peripheral event rather than the completion of some calculation.
2) If needed (from the viewpoint of model fidelity), an extensive period of CPU activity can be modeled in SMURPH by appropriately (and explicitly) delaying certain state transitions.

In most cases, we can ignore the fact that the execution of a PicOS program takes time at all and only focus on reflecting the accurate behavior of the external events. With this assumption, the job of porting a PicOS praxis to its VUE$^2$ model can be made simple. To further increase the practical value of such a model, SMURPH provides for the so-called *visualization mode* of execution. In that mode, SMURPH tries to map the virtual time of modeled events to real time, such that the user has an impression of talking to a real application. This is only possible if the network size and complexity allow the simulator to catch up with the model execution to real time; otherwise, a suitable slow motion factor can be applied.

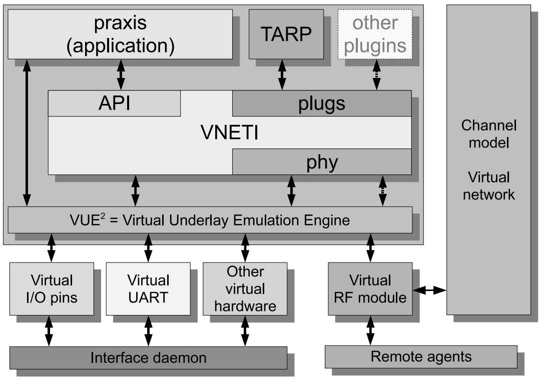


Fig. 3: The structure of a VUE$^2$ model.

### B. Model scope

SMURPH threads are programmed in C++, which we have extended with new keywords and constructs. A special preprocessor (dubbed SMPP) processes the SMURPH source to produce pure C++ code. PicOS praxes are programmed in plain C with the assistance of a few macros (see Fig. 1) expanded by the standard C preprocessor. Putting trivial syntactic issues aside, the most fundamental difference between the two systems is the fact that a SMURPH model must describe the whole network (i.e., a multitude of nodes, each of them running a private copy of the application), while a complete PicOS praxis is a single program that runs on a single device. This difference becomes more pronounced if the network consists of nodes running different praxes, a not uncommon scenario.

We make extensive use of C++ classes to accomplish the conversion from a single-application node to a multi-node (and possibly multi-application) emulator. In the conversion, each PicOS praxis becomes a C++ class. Most of the praxis functions and variables become member functions and attributes of the class, respectively. For truly global (node indifferent) functions and data, the compiler need not associate them with a specific class and can instead keep them global. When the emulator executes and builds the network, it represents each node as an object (i.e., instance of the appropriate class).

Beyond the “adaptation layer” for PicOS praxes, the VUE$^2$ extension to SMURPH implements detailed models for the physical hardware (Section III-C). In terms of communication, SMURPH brings in a powerful generic wireless channel model [22] that provides enough flexibility to implement arbitrarily complex propagation models. All of this potential for modeling allows us to confidently and comprehensively verify applications before uploading the code to physical nodes.

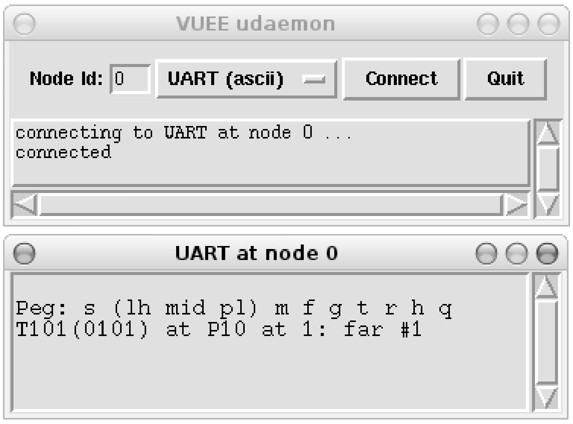
### C. Peripherals

The current version of VUE$^2$ implements detailed models for a significant subset of PicOS-supported peripherals that include serial communications (UART), physical sensors, general-purpose I/O (GPIO), digital-to-analog converters (DACs), analog-to-digital converters (ADCs), and light-emitting diodes (LEDs). Some of these devices, such as the GPIO pins, may require input or produce output. For such devices, VUE$^2$ offers a variety of peripheral-dependent options. In the case of the GPIO pins, the developer can describe their I/O via (a) the initial network description (for input only), (b) external files (to pre-generate/script input and log output), or (c) communication over a network socket. In the third case, VUE$^2$ provides a special program named `udaemon` that allows the developer to interactively read from and write to the peripheral.

The `udaemon` application is a fundamental component in the *interactive* emulation of a wireless sensor node and its peripherals. The initial window in the GUI (Fig. 4, top) allows the user to open peripheral-specific windows for individual emulated nodes. For example, the UART window (Fig. 4, bottom) allows two-way communication with a node over a virtual serial interface. This `udaemon` application provides access to all VUE$^2$-modeled peripherals.

## IV. Application Development

We have used both PicOS and VUE$^2$ together to implement and test a variety of practical wireless network applications such as passively monitoring environmental conditions and actively tracking the movement of indoor objects. In the subsections that follow, we introduce a few of our applications and highlight VUE$^2$’s ability to accommodate their specific (often peripheral-related) virtualization requirements.



```
connecting to UART at node 0 ...
connected
```

```
Peg: s (lh mid pl) m f g t r h q
T101(0101) at P10 at 1: far #1
```

Fig. 4: A screen shot of `udaemon` showing its primary window (top) and interaction with an individual node over UART (bottom).

### A. EcoNet

The *EcoNet* project, conceived with the Earth Observation Systems Laboratory at the University of Alberta, aims to monitor an environment’s sunlight, temperature, and humidity. Wireless sensor nodes distributed throughout an environment periodically measure these characteristics and then report them to a sink node. In this scenario, a single deployment uses two separate applications: a *collector* to read/transmit sensor values and an *aggregator* to receive sensor values. The collector application uses the PicOS sensor functionality to read the current values from its analog sensors.

During execution, the collector application calls the PicOS function `read_sensor` to obtain the latest values from a device’s sensors. When running on the hardware, PicOS obtains these values using the hardware’s ADC. In the emulator, the user can provide the sensor values graphically on a per-node basis using slider widgets. During an experiment, simply dragging the sliders interactively changes the sensor values visible at a node.

### B. Mousetraps

The *Mousetrap* project, conceived with researchers in the University of Alberta’s biology department, aims to monitor the common live-catching trap. When a rodent enters one of our traps, the movement of the ramp triggers a physical switch that we have added to the trap. The triggering of the switch creates a message that the node sends to its associated sink. In the above network, nodes run a single application that uses PicOS’s pin monitoring/notifier functionality for digital input.

The API for the pin notifier functionality includes functions to enable, disable, and check it. An application that uses it will wait for the predefined event PMON_NOTEVENT. On the actual hardware, this event is implemented through interrupt handlers and some auxiliary functions, e.g., needed for debouncing the switch. In the emulator, we use essentially the same code, and the user can graphically change the value of a monitored input pin using a button in the GUI.

### C. *Tags and Pegs*

The *Tags and Pegs* project at Olsonet aims to locate a sensor-enabled object within a sensor-enabled environment. Nodes in the network periodically broadcast short messages, and then other nodes use received signal strengths to perform localization. This deployment also uses two applications: one for mobile nodes and one for static nodes. To obtain signal strength values, nodes use PicOS's standard packet reception functions.

When the application calls the PicOS function net_rx, the application can retrieve a received packet from VNETI, and at the same time, the corresponding signal strength. In the hardware, the signal strength comes from the radio transceiver. For the emulator, SMURPH's generic wireless model calculates signal strengths for all virtual receptions, and the virtualization of the PicOS API uses these calculated values. From the application's perspective, there is no difference between the hardware and virtual environment.

## V. Case Study: Ping

In this section, we describe a simple *ping* application to illustrate how our platform transforms a single set of source files into code suitable for compilation with both the hardware (PicOS) and the emulator (VUE$^2$). In this example, two nodes run identical copies of the software.[*] When powered on, a node immediately begins to broadcast unaddressed ping packets that contain a locally maintained sequence number. Whenever a node receives such a ping packet, it broadcasts an acknowledgment that contains the received sequence number. Upon receiving an acknowledgement with the last sent sequence number, a node increments its locally maintained sequence number and immediately broadcasts another ping packet. If a ping packet goes unacknowledged, a node retransmits the ping after a predetermined delay.

Before presenting code, the term *thread* (used loosely in Section II) requires some clarification. In the context of the source code, our platform makes a distinction between processes with and without arguments. By doing so, we can improve compatibility between the hardware and emulator targets. We call a process that expects a typed data argument on initialization a strand. Many instances of a strand may exist at any given time, where each operates on its own local (private) data. We call a process that tends to operate on global data and does not accept such an argument a thread. Only one instance of a thread may exist at any given time.

We logically divide the ping application into three processes. In our platform, execution begins at the *thread* root (Fig. 5a, right), which is akin to the function main in a traditional C program. In a *strand* named sender (Fig. 5a, left), we place the code that transmits ping packets. We use a strand so that we can pass it the retransmission delay as an argument (in this case, line 34 sets the delay to about two seconds). Finally, we place the code for packet reception and acknowledgement generation in a *thread* named receiver (not shown).

Consider first the thread named root (Fig. 5a, right). The keyword entry identifies a state boundary and its argument identifies the state name. This thread contains a single state named RS_INIT. The first three lines of this state essentially serve as a constructor to (a) register a physical device with VNETI, (b) register a protocol plug-in with VNETI, and (c) open a session using that device and protocol. After some error handling code, this thread continues to enable the radio's transmitter and receiver along with starting the previously introduced processes sender and receiver.

It is quite possible for a single application to support a variety of different physical radio transceivers. Such a case might arise where a particular deployment's specific characteristics later dictate the best hardware. VNETI's abstractions make this type of flexibility possible. For each supported radio transceiver, the VNETI API provides a single function with the prefix phys_ to register the physical device (e.g., Fig. 5a:25). To support multiple radios, developers can use trivial preprocessor directives (e.g., #if and #endif) to call the appropriate phys_ function. Beyond this initialization stage, most applications require no further changes to switch between different transceivers.

Another noteworthy point is VNETI's protocol plug-in registration using the function tcv_plug(...) (e.g., Fig. 5a:26). By registering the *null* protocol plug-in for the ping application, calls to functions in the VNETI API provide the programmer with a more or less direct connection to the network. The programmer then has much flexibility to manage the packet overhead. Beyond the *null* plug-in, our platform also implements the Tiny Ad hoc Routing Protocol (TARP) to perform the ad hoc routing that we mentioned in our introduction. Given our plug-in oriented approach, users can implement further protocols as desired.

The remaining VNETI API functions begin with the prefix tcv_ and primarily serve as state and buffer management. Table I briefly describes some of the tcv_ functions relevant to the presented code.

In Fig. 5a, left, we present the code for the strand sender. In our platform, we define a number of data types to provide consistent variable sizes between the different targets. The keyword word that appears in the definition of sender identifies the type of its *data* argument (the word type is a 16-bit unsigned value). Later in the strand, the user can access this data argument using the (implicit) variable data. Upon entering the state SN_SEND, code checks whether the node received an acknowledgement for the last ping. If so, it immediately proceeds to send another ping. If not, it (a) sets a timer to delay before rebroadcasting the ping and (b) waits for

[*] The application's complete source code is available online at http://tinyurl.com/67a4t6.

```
strand (sender, word)
  entry (SN_SEND)
    if (last_ack != last_snt) {
      delay ((word)data, SN_NEXT);
      when (&last_ack, SN_SEND);
      release;
    }
    last_snt++;
    proceed (SN_NEXT);
  entry (SN_NEXT)
    x_packet = tcv_wnp (SN_NEXT, sfd,
                        DATA_LENGTH);
    x_packet[0] = 0;
    x_packet[1] = PKT_DAT;
    ((lword*)x_packet)[1] = wtonl (last_snt);
    tcv_endp (x_packet);
  entry (SN_OUT)
    ser_outf (SN_OUT, "SND %lu, len = %d\r\n",
              last_snt, DATA_LENGTH);
    proceed (SN_SEND);
endstrand
```

```
thread (root)
  entry (RS_INIT)
    // setup the radio
    phys_dm2200 (DEV_ID, MAX_LENGTH);
    tcv_plug (DEV_ID, &plug_null);
    sfd = tcv_open (WNONE, DEV_ID, 0);
    if (sfd < 0) {
      diag ("Cannot open tcv interface");
      halt ();
    }
    // start sender
    tcv_control (sfd, PHYSOPT_TXON, NULL);
    runstrand (sender, 2048);
    // start receiver
    tcv_control (sfd, PHYSOPT_RXON, NULL);
    runthread (receiver);
    // done with initialization
    finish;
endthread
```

(a) User-written code for the processes **sender** and **root** prior to preprocessing.

```
int sender (word zz_st, address zz_da) {
  word *data = (word*) zz_da;
  switch (zz_st) {
  case 0:
    if (last_ack != last_snt) {
      delay ((word)data, 10);
      zzz_uwait ((word)(&last_ack),0);
      zz_restart_entry ();
    }
    last_snt++;
    proceed (10);
  case 10:
    x_packet = tcv_wnp (10, sfd, 10);
    x_packet[0] = 0;
    x_packet[1] = 0xABCD;
```

```
    ((lword*)x_packet)[1] =
     ((((last_snt) & 0xffff) << 16) |
     (((last_snt) >> 16) & 0xffff));
    tcv_endp (x_packet);
  case 20:
    ser_outf (20, "SND %lu, len = %d\r\n",
              last_snt, 10);
    proceed (0);
    break;
  default:
    if (zz_st == 0xffff)
      return (0);
    zz_badstate ();
  }
  return 1;
}
```

(b) Preprocessed code for the process **sender** when targeting **PicOS**.

```
void sender::zz_code () {
  switch (TheState) {
  case SN_SEND: __state_label_SN_SEND:
    if ((((PingNode *)TheStation)-> _na_last_ack) != (((PingNode *)TheStation)-> _na_last_snt)) {
      ( ((PicOSNode*)TheStation)->_na_delay ((word)data,SN_NEXT) );
      ( ((PicOSNode*)TheStation)->_na_when (
         ((int)(IPointer)(&(((PingNode *)TheStation)-> _na_last_ack))), SN_SEND) );
      return;
    }
    (((PingNode *)TheStation)-> _na_last_snt)++;
    do { zz_AI_timer.zz_proceed (SN_NEXT); return; } while (0);
  case SN_NEXT: __state_label_SN_NEXT:
    x_packet = ( ((PicOSNode*)TheStation)->_na_tcv_wnp (
                  SN_NEXT,(((PingNode *)TheStation)-> _na_sfd),10) );
    x_packet[0] = 0;
    x_packet[1] = 0xABCD;
    ((lword*)x_packet)[1] = ((((((PingNode *)TheStation)-> _na_last_snt)) & 0xffff) << 16) |
                             (((((PingNode *)TheStation)-> _na_last_snt)) >> 16) & 0xffff));
    ( ((PicOSNode*)TheStation)->_na_tcv_endp (x_packet) );
  case SN_OUT: __state_label_SN_OUT:
    ( ((PicOSNode*)TheStation)->_na_ser_outf (SN_OUT, "SND %lu, len = %d\r\n",
                                              (((PingNode *)TheStation)-> _na_last_snt), 10) );
    do { zz_AI_timer.zz_proceed (SN_SEND); return; } while (0);
  }
}
```

(c) Preprocessed code for the process **sender** when targeting **VUE$^2$**.

Fig. 5: Excerpts from the ping application's source code both before and after preprocessing.

TABLE I: Some of the most common VNETI API functions and their descriptions.

| Function | Description |
|---|---|
| `tcv_plug` | Configures a protocol plug-in for the network interface; in the ping application, the null plug-in provides a more or less direct connection to the network. |
| `tcv_open` | Opens a session and returns a session descriptor (akin to a file descriptor). |
| `tcv_control` | Allows the application to change various parameters associated with the transceiver; in the ping application, we use it to enable the transmit and receive functionality of the radio. |
| `tcv_rnp` | Acquires the next packet queued for reception at the session. |
| `tcv_wnp` | Requests a packet handle from VNETI in order to send a new outgoing packet. |
| `tcv_left` | Determines the length of a packet acquired by `tcv_rnp`. |
| `tcv_endp` | Indicates explicitly the moment when a packet has been processed and is no longer needed. |

a signal (IPC) on the address of `last_ack`. The `receiver` process (not shown) triggers the address of `last_ack` when it receives an expected acknowledgement. By waiting for this signal in `sender`, the application can then immediately advance the sequence number and send out a new ping packet.

The programmer's effort amounts to writing code similar to that presented in Fig. 5a. Note that the context for this code is a single node and it is plain C code, albeit enhanced with new "keywords" to improve clarity and simplify programming that we have implemented as preprocessor macros. Since the code is plain C, compiling for PicOS simply uses the standard C preprocessor and compiler. For VUE$^2$, a specialized preprocessor makes the more complicated transition to C++ code where multiple applications and nodes must operate in a single simulation environment that preserves the state of those individual nodes.

### A. Preprocessing

In Fig. 5b and c, we show the changes made to `sender` by the respective preprocessor to prepare the code for compilation with PicOS and VUE$^2$. In this subsection, we describe some of the changes along with the reasoning behind them.

The first thing to notice is that in both cases the code's general structure remains the same. The user-written finite state machine using our `strand`/`entry` "keywords" becomes a switch on a variable containing the current state. In the PicOS case, the preprocessor substitutes state names with integer constants because the labels are `#define` preprocessor directives. For VUE$^2$, the symbolic names remain because an enumerated type represents states and thus the compiler introduces the integer constants rather than the preprocessor.

In the VUE$^2$ code, notice the appearance of several new variables (i.e., `TheStation` and `TheState`). These variables (and others) arise in the transition from a single-node (hardware) environment to a multi-node (simulated) environment. The simulator represents each node in the network as an object, and the variable `TheStation` points to the node currently being simulated. Another global variable named `TheProcess` identifies the current process (e.g., `sender`) within that node that the simulator is evaluating. Finally, a global variable named `TheState` holds an integer that identifies the current state within that process. At any point, these three variables collectively describe the current state of the simulation.

Notice that all accesses to system calls and node-specific variables within the user's application use the pointer `TheStation` (e.g., Fig. 5c:V04-V07,V10). At different places in the preprocessed code, the single object is typecast to either a `PingNode` or a `PicOSNode`. The derivation of the relevant classes is as follows. SMURPH provides a base class to represent a piece of hardware running in the network named `Station`. VUE$^2$ then introduces the notion of a `PicOSNode` as a specialized type of station and thus derives it from `Station`. At this level, VUE$^2$ defines the PicOS system calls (including those for VNETI) and internal state variables. From here, VUE$^2$ derives a further class that is protocol plug-in specific; it contains the functions and variables necessary to implement the plug-in. In this case, we use the *null* protocol plug-in and thus this further subclass is of the type `NNode`. Note that VUE$^2$ also provides a class `TNode` for nodes that run the *TARP* protocol plug-in. Finally, the class representing the actual application (in this case `PingNode`) inherits from the protocol-specific class (in this case `NNode`) and further defines the processes and variables of the user's application. When starting a simulation, VUE$^2$ builds all of the network nodes using the lowest-level (and most complete class), which in this case is `PingNode`. To see the different typecasts, first look to line V04, which typecasts to `PingNode` when accessing node-private application variables, and then to lines V05 and V06, which typecast to `PicOSNode` for making system calls.

Notice that the C++ version contains additional labels on lines V03, V12, and V20 that begin with the text `__state_label_`. When the simulator comes across the keyword `proceed` (e.g., lines V11 and V23), it saves the next state in the variable `TheState` and then returns control to the scheduler to make the state transition. The scheduler may not immediately return control to the process if other events occur at the same time. When the process does resume, the `switch` statement on the variable `TheState` moves the process into the appropriate state. Sometimes, SMURPH users will want to make a transition that does not involve the scheduler. In these cases, using the command `sameas` (not shown) rather than `proceed` accommodates an immediate transition using the label `__state_label_` along with a `goto` statement. Note that none of the code written for VUE$^2$/PicOS can currently make use of the `sameas` functionality.

Readers may be unfamiliar with the construct

```
do  ... while (0);
```

introduced on lines V11 and V23. It results from a macro expansion of our keyword `proceed`. Essentially, this code is a C idiom to define a macro consisting of multiple statements that has the syntactic rights of a single statement.

In both the PicOS and VUE$^2$ preprocessed code, name mangling occurs. In PicOS, preprocessing appends the characters `zzz_`, `zz_`, or `x_` to functions and variables internal to PicOS in an attempt to avoid conflicts. In VUE$^2$, some similar mangling occurs plus further mangling on node-private variables where the processor appends `_na_` for similar reasons.

## VI. Conclusion

In this paper, we described our "holistic" platform for building wireless ad hoc sensor networks and focused on its most representative and essential component: VUE$^2$ (the Virtual Underlay Emulation Engine). Using it, developers can write applications in C, rather than a new programming language or bytecode, and then easily target to both hardware nodes and our emulator.

Through the development of several applications, we have found that the finite state machine paradigm allows for the natural representation reactive applications. By using VUE$^2$ during the development stage, we can test our applications exhaustively in a virtual environment before investing time to program physical hardware. When we later move our applications to the hardware, they perform within our expectations.

## Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) and the Informatics Circle of Research Excellence (iCORE) for helping fund this research.

## References

[1] C. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance Vector routing (DSDV) for mobile computers," in *Proc. of SIGCOMM'94*, Aug. 1993, pp. 234–244.

[2] T.-W. Chen and M. Gerla, "Global state routing: a new routing scheme for ad-hoc wireless networks," in *Proc. of ICC'98*, June 1998.

[3] V. Park and M. Corson, "A performance comparison of TORA and ideal link state routing," in *Proc. of IEEE Symposium on Comp. and Comm.*, June 1998.

[4] J. Li, J. Jannotti, D. D. Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. of the ACM/IEEE Intl. Conference on Mobile Computing and Networking (MOBICOM' 00)*, 2000, pp. 120–130.

[5] C. Perkins, E. B. Royer, and S. Das, "Ad-hoc On-demand Distance Vector routing (AODV)," February 2003, Internet Draft: draft-ietf-manet-aodv-13.txt.

[6] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

[7] T. Henderson, J. Park, N. Smith, and R. Wright, "From motes to Java stamps: Smart sensor network testbeds," in *Intelligent Robots and Systems*, Las Vegas, NV, Oct. 2003, pp. 799–804.

[8] P. Levis and D. Culler, "Maté: A tiny virtual machine for sensor networks," in *Proc. of the 10th Intl. Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, CA, Oct. 2002, pp. 85–95.

[9] [Online]. Available: http://www.dalvikvm.com/

[10] [Online]. Available: http://code.google.com/android/

[11] E. Akhmetshina, P. Gburzyński, and F. Vizeacoumar, "PicOS: A tiny operating system for extremely small embedded platforms," in *Proc. of ESA'03*, Las Vegas, Jun. 2003, pp. 116–122.

[12] O. Dahl and K. Nygaard, "Simula: A language for programming and description of discrete event systems," Norwegian Computing Center, Oslo, Introduction and user's manual, 5th edition, 1967.

[13] G. Birthwistle, O. Dahl, B. Myhrhaug, and K. Nygaard, *Simula Begin*. Oslo: Studentlitteratur, 1973.

[14] A. Gustafsson, "Threads without the pain," *Social Computing*, vol. 3, no. 9, pp. 34–41, 2005.

[15] D. Thomas, C. Fowler, and A. Hunt, *Programming Ruby: The Pragmatic Programmer's Guide*. The Pragmatic Programmers, 2004, second edition.

[16] P. Gburzyński, *Protocol Design for Local and Metropolitan Area Networks*. Prentice-Hall, 1996.

[17] W. Dobosiewicz and P. Gburzyński, "Protocol design in SMURPH," in *State-of-the-art in Performance Modeling and Simulation*, J. Walrand and K. Bagchi, Eds. Gordon and Breach, 1997, pp. 255–274.

[18] K. Altisen, F. Maraninchi, and D. Stauch, "Aspect-oriented programming for reactive systems: a proposal in the synchronous framework," Verimag CNRS, Research Report #TR-2005-18, Nov. 2005.

[19] B. Yartsev, G. Korneev, A. Shalyto, and V. Ktov, "Automata-based programming of the reactive multi-agent control systems," in *Intl. Conference on Integration of Knowledge Intensive Multi-Agent Systems*, Waltham, MA, Apr. 2005, pp. 449–453.

[20] W. Olesinski, A. Rahman, and P. Gburzyński, "TARP: a tiny ad-hoc routing protocol for wireless networks," in *Australian Telecommunication, Networks and Applications Conference (ATNAC)*, Melbourne, Australia, Dec. 2003.

[21] P. Gburzyński, B. Kaminska, and W. Olesinski, "A tiny and efficient wireless ad-hoc protocol for low-cost sensor networks," in *Proc. of Design Automation and Test in Europe (DATE'07)*, Nice, France, Apr. 2007, pp. 1562–1567.

[22] P. Gburzyński and I. Nikolaidis, "Wireless network simulation extensions in SMURPH/SIDE," in *Proc. of the 2006 Winter Simulation Conference (WSC'06)*, Monetery, California, Dec. 2006.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Wireless Ad hoc Networks: Where Security, Real-time and Lifetime Meet

Zdravko Karakehayov

Technical University of Sofia,
Bulgaria
Department Computer Systems

Email: zgk@tu-sofia.bg

*Abstract*—**This paper deals with the multihop nature of wireless ad hoc networks where security, real-time and lifetime meet. We propose a hierarchical communication model to study how medium access control and network energy signatures provide opportunities for power-efficient partitioning of communication links. We show possibilities to integrate multihop services into routing protocols. At the same time, malicious nodes included in a routing path may misbehave and the paper analyzes vulnerable points in case of communication attacks. The REWARD algorithm for secure routing is used as a main example for corrective actions. Nodes listen to neighbor transmissions to detect black hole attacks. The energy overhead is broken down into static, the additional energy required to watch for attacks, and dynamic, which is application specific. We evaluate the static energy overhead associated with symmetrical routing.**

## I. Introduction

AD HOC networks have a wide spectrum of military and commercial applications. Ad hoc networks are employed in situations where installing an infrastructure is too expensive, too vulnerable or the network is transient. The interaction between the nodes is based on wireless communication. Packets are forwarded in a multihop manner. Nodes have a limited radio footprint and when a node receives a packet it applies a routing algorithm to select a neighbor for forwarding.

There is a class ad hoc networks, sensor networks, where the requirements for lifetime and size of the nodes are driven to extremes. A wireless sensor network consists of a large number of nodes that may be randomly and densely deployed. Sensor nodes are capable of sensing many types of information such as temperature, light, humidity and radiation. Sensor networks must collect data in an area of interest for months or years. Since the energy is a scarce and usually non-renewable resource, the network's functionality must be viewed from a low-power perspective. Sensor network nodes execute three major tasks: sensing, computation and communication.

Communication energy dominates the overall energy budget. The greater than linear relationship between transmit energy and distance promises to reduce the energy cost when the radio link is partitioned. Nodes calculate the distance and tune their transmit power accordingly. Consequently, it would be beneficial to use several hops to reach a node within the transmission radius instead of a direct link. Along with available locations of the nodes, a multihop optimization requires an appropriate power model. For some applications it is not necessary nodes to have real coordinates. Instead, nodes may have virtual coordinates: hop-distances to other nodes.

Moreover, some applications require the network to influence the environment via actuators. Synchronization between input and output demands real-time traffic. Real-time forwarding of packets under multihop communication scheme is a serious challenge. When we factor in security, the outlook becomes even more grim. Packets travel over several nodes and malicious attacks are easy to organize. To detect malicious influence and wage corrective actions the nodes must spend extra energy. Consequently, the multihop nature of ad hoc networks, while beneficial for energy reduction, brings the packets delivery time up. The dynamic nature of the network and the power-efficient partitioning of communication links in particular, often result in unpredictable traffic timing parameters. Enemy nodes included in a routing path may misbehave and any attempt to make the network less vulnerable requires extra energy and affects the lifetime, thus closing the loop.

## II. Related Work

Different MAC protocols are discussed in [1]-[6]. Energy efficiency is the primary goal of the research. While a power saving technique, termed Span [1], dynamically splits the nodes into sleeping nodes and forwarding nodes, S-MAC, a medium access control protocol [2], establishes a low duty cycle operation in all nodes. ExOR, extremely opportunistic routing is a routing method developed to reduce the total number of transmissions taking into account the actual packet propagation [3]. DTA, data transmission algebra, has been developed to generate complex transmission schedules based on collision-free concurrent data transmissions [5]. In related research we proposed ALS-MAC, a medium access control protocol where contention-based advertising slots are mapped to scheduled-based transmission slots [6]. The energy model employed in this paper has been adopted from [7], [8]. Despite there being a plethora of sensing and MAC papers, comparatively little has been published on the companion task of actuation and real-time requirements. Sensor-actuator networks are discussed in [9], [10]. The problem of obtaining virtual coordinates is addressed in [11].

978-83-60810-14-9/08/$25.00 © 2008 IEEE

Different aspects of node architectures and capabilities can be found in [12]-[17]. The power reduction methods discussed in [15]-[17] are not confined to computation energy of network nodes. They can be applied, also, in other cases where voltage-scalable or speed-scalable CPUs follow the current requirements and save energy. Another approach to reduce the power consumption is to remove hardware used for localization, such as GPS, and utilize receive signal strength, RSS. The resulting accuracy and impact factors are investigated in [14].

Methods for energy efficient multihop communication are discussed in [18]-[22]. A detailed investigation for simple settings is available in [19]. In related research we studied multihop optimization for non-regular topologies [6], [10]. An Aloha type access control mechanism for large, multihop, wireless networks is defined in [21]. The protocol optimizes the product of the number of simultaneously successful transmissions per unit of space, spatial reuse, by the average range of each transmission.

A review of routing protocols for wireless ad hoc networks is available in [23]. The problem of radio irregularity is discussed in [24]. Later in Section V, we compare distances with the communication range. Due to radio irregularity some neighbors located within the transmission disk may be inaccessible while some remote nodes, outside the disk, will be capable to communicate. Since quite a few processor architectures vie for attention in the realm of sensor networks, target-aware modeling of routing algorithms helps to evaluate important timing properties [25]. Security of wireless sensor networks is in focus in [26]-[31]. Two papers, [22] and [30], emphasize the fact that multiobjective design is needed. Listening to neighbor transmissions to detect black hole attacks is discussed in [32]-[35].

## III. Communication Model

The communication model describes a packet forwarding from a source to a destination. The destination is within the communication range of the source. The communication model, C, has three components: a set of the locations of nodes, L, a medium access control model, M, and an energy model, E.

$$C=\{L, \mathrm{M}, E\} \tag{1}$$

### A.Medium access control model

Medium access control (MAC) mechanism has a significant impact on the energy efficiency [2], [4], [6]. Currently available MAC protocols for wireless sensor networks can be broken down into two major types: contention-based and scheduled-based. While under contention-based protocols nodes compete among each other for channel access, scheduled-based schemes rely on prearranged collision-free links between nodes. There are different methods to assign collision-free links to each node. Links may be assigned as time slots, frequency bands, or spread spectrum codes. However, size and cost constrains may not permit allocating complex radio subsystems for the node architecture. Logically, TDMA scheduling is the most common scheme for the domain of wireless sensor networks. The limited communication range of network nodes provides an extra opportunity for collision-free interaction, space division access [5], [6], [21].

- *Assume scheduled links*

In order to save energy nodes should stay in a sleeping mode as long as possible. Ideally, nodes should have prearranged collision-free links and wake up only to exchange packets. This MAC approach can be termed Assume Scheduled Links, **ASL**. The **ASL** model has two parameters: a packet length in bits, p, and a bit rate, B.

$$\mathrm{M}=\{\mathrm{ASL}, \mathrm{p}, \mathrm{B}\} \tag{2}$$

While **ASL** is a theoretical concept, it helps to outline the floor of the energy required for communication.

- *Beacon Advertise Transmit*

Beacon Advertise Transmit, **BAT**, model is a widespread MAC mechanism [4]. Beacons are employed to synchronize internode communications. A beacon period, $T_B$ , includes two major sections. The period begins with a traffic indication window, $T_A$ . During $T_A$ all nodes are listening and pending packets are advertised. The nodes addressed till the end of $T_A$ send acknowledgements and receive data packets. Data transmissions are followed by acknowledgement frames to confirm successful reception. Fig. 1 illustrates a beacon period.

The **BAT** model has five parameters: $T_A$ , $T_B$ , a data packet length in bits, p, a control packet length in bits, q, and a bit rate, B.

$$M=\{\mathrm{BAT}, \mathrm{T}_A, \mathrm{T}_B, \mathrm{p}, \mathrm{q}, \mathrm{B}\} \tag{3}$$

### B. Energy model

The energy used to send a bit over a distance *d* via radio communication may be written as

$$\mathrm{E}=ad^n+b \tag{4}$$

where *a* is a proportionality constant [7], [8]. The radio parameter *n* is a path loss exponent that describes the rate at which the transmitted power decays with increasing distance. Typically, *n* is between 2 and 4. The *b* constant is associated with specific receivers, CPUs and computational algorithms. Thus the model emerges as

$$E=\{a, n, b, \mathrm{P}_R\} \tag{5}$$

where $P_R$ is the power consumption of a turned on receiver.

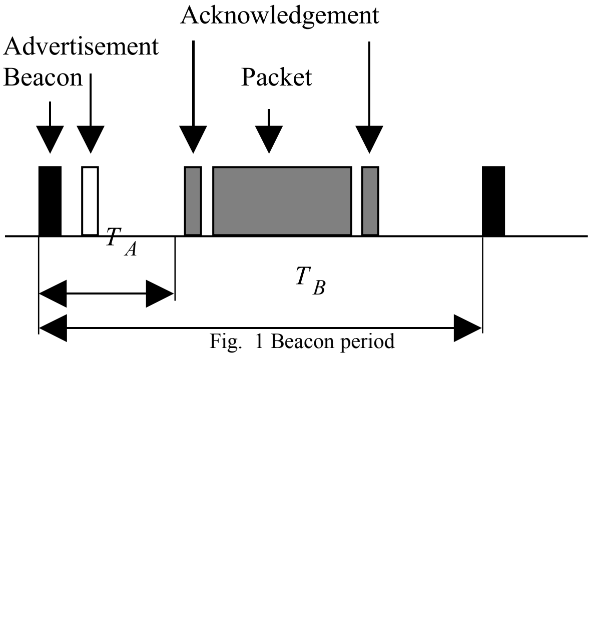


Fig. 1 Beacon period

## IV. Real-Time Behavior

Using the BAT model and counting the beacon periods nodes are in position to calculate the packets delivery time. While this completely applies for destination nodes, intermediate nodes can use the actual packet propagation time and virtual coordinates to foresee the overall delivery time.

In the large, energy versus real-time tradeoffs can be resolved via different values assigned for the beacon period. In the small, at each hop nodes decide whether to include an extra intermediate node for power efficiency or to forward the packet as fast as possible. The local decision is based on the actual propagation of the packet measured in number of beacon periods and the remaining number of hops.

## V. Lifetime

An ad hoc network lifetime can be measured by the time when the first node runs out of energy, or a network can be declared dead when a certain fraction of nodes die. Alternatively, the system lifetime can be measured by application-specific parameters, such as the time until the system can no longer provide acceptable quality of service. Clearly, the higher the energy efficiency is, the longer the network will survive. The energy efficiency can be optimized at three levels.

### *A. Node architecture*

A typical node is built around a low-power microcontroller [12],[13],[15]. Wireless transceivers create physical links between nodes. Hardware provides the following low-power mechanisms. The receiver and transmitter can be individually enabled and disabled. The transmit power can be adjusted gradually. For many applications nodes are capable of determining their coordinates. Voltage-scalable systems may apply dynamic voltage or clock frequency scaling to reduce the power consumption.

### *B. Multihop routing service*

Once the routing protocol has provided the next relay another neighbor can be considered to partition the link. The number of hops is increased to save energy. As an additional benefit, the reduced transmit power allows better spatial reuse. Fig. 2 shows how an intermediate node can be used to break down the link between a source S and a destination D into two hops.

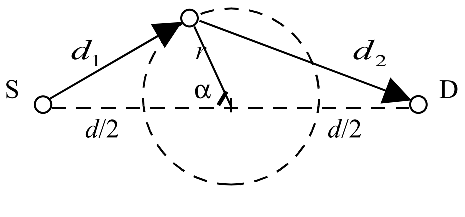


Fig. 2 Routing via an intermediate node

**Theorem 1**. Let $C=\{\{\text{ASL,p,B}\},\{a,4,b,\text{P}_R\}\}$ be the communication model of a wireless ad hoc network. If the distance between the source S and the destination D $d \geq \left((8\text{b}+(\text{p/B})P_R)/7\text{a}\right)^{1/4}$ and the distance between an intermediate node and the halfway point between S and D

$$r \leq (-0.75d^2+0.25(9\text{d}^4-a^{-1}(8\text{b}-7ad^4+(\frac{\text{p}}{\text{B}})P_R))^{\frac{1}{2}})^{\frac{1}{2}},$$

the two-hop communication requires less energy than the direct link.

**Proof.** We must prove when the following inequality holds.

$$ad_1^4+b+ad_2^4+b+2(\text{p/B})P_R \leq ad^4+b+(\text{p/B})P_R \quad (6)$$

Taking into account that

$$d_1=\left(d^2/4-dr\cos\alpha+r^2\right)^{1/2} \quad (7)$$

$$d_2=\left(d^2/4+dr\cos\alpha+r^2\right)^{1/2} \quad (8)$$

We get

$$16ar^4+8ad^2\left(1+2\cos^2\alpha\right)r^2+8\text{b}-7ad^4+(\text{p/B})P_R \leq 0 \quad (9)$$

The inequality has solutions if and only if $d \geq \left((8\text{b}+(\text{p/B})P_R)/7\text{a}\right)^{1/4}$ . Since the threshold value for the distance $r$ will vary with α, we take the worst case, $\cos\alpha=1$ .

Using the quadratic formula,

$$r \leq (-0.75d^2+0.25(9\text{d}^4-a^{-1}(8\text{b}-7ad^4+(\frac{\text{p}}{\text{B}})P_R))^{\frac{1}{2}})^{\frac{1}{2}} \quad (10)$$

□

Fig. 3 shows plots for the radius $r$ compared with half of the distance. This example assumes two bit rates, 1 Mbps and 0.5 Mbps, $a$ = 0.2 fJ/m$^4$, $b$ = 1 nJ, $\text{P}_\text{R}$= 10 mW and p = 128 bit.

**Theorem 2**. Let $C=\{\{\text{BAT,T}_A\text{,T}_B\text{,p,q,B}\},\{a,4,b,\text{P}_R\}\}$ be the communication model of a wireless ad hoc network. Let the average number of neighbors listening to a beacon transmission be D. If the distance between the source S and the destination D

$$d \geq ((b(3\text{q}+p)+P_R B^{-1}(q+p)$$
$$+P_R\text{DT}_A)a^{-1}(3.5625q+0.875\text{p})^{-1})^{1/4}$$

and the distance between the intermediate node and the halfway point between S and D

$$r \leq (-0.25d^2(10.5\text{q}+3\text{p}+0.5\text{q}d)(3\text{q}+p+qd)^{-1}$$
$$+0.5\text{a}^{-1}(3\text{q}+p+qd)^{-1}(0.25a^2d^2(10.5\text{q}+3\text{p}+0.5\text{q}d)^2$$
$$-2\text{a}(3\text{q}+p+qd)(-ad^4(3.5625q+0.875p)$$
$$+b(3\text{q}+p)+P_R B^{-1}(q+p)+P_R\text{DT}_A))^{1/2})^{1/2}$$

the two-hop communication requires less energy than the direct link.

□

The radius $r$ for a given distance $d$ indicates application-specific opportunities for power-efficient partitioning of communication links. Fig. 4 compares **ASL** and **BAT** MAC models for a bit rate of 512 Kbps.

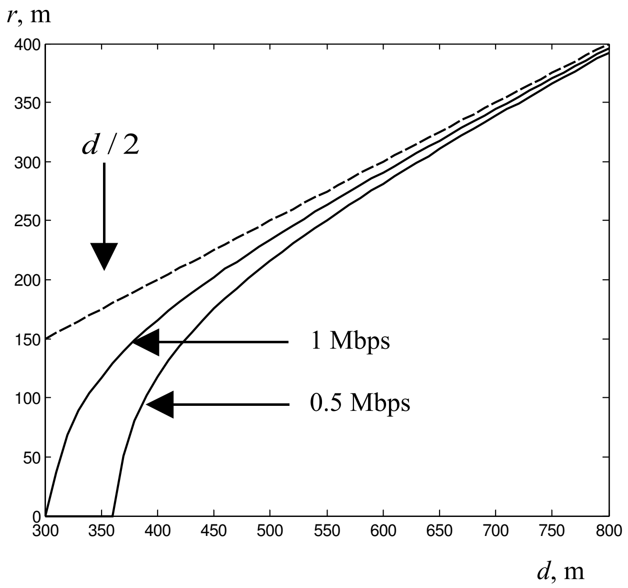


Fig. 3 Radius *r* scales with the distance for two bit rates

### *C.Routing algorithms*

Routing algorithms can be based on two major approaches: topology-based and position-based routing [23]. The topology-based algorithms can be further split into table-driven and demand-driven. The main idea behind the table-driven routing protocols is to create a clear picture of all available routes from each node to every other node in the network. In contrast to the table-driven protocols, the demand-driven algorithms create routes via route discovery procedures only when a necessity arises.

Position-based routing algorithms utilize the physical positions of the participating nodes [19],[21],[23]. Position-based or geographic routing does not require each node to have the locations of all other nodes. Each node keeps track of the coordinates of its neighbors and their neighbors. A greedy routing algorithm based on geographic distance selects the closest to the destination neighbor for the next hop [19].

Assume that the nodes of a wireless ad hoc network are members of the following set $N = \{N_1, N_2, N_3, \dots, N_{n(N)}\}$. The nodes are placed in a rectangular region of *X* by *Y*. The distance between node i and node j is *d*(i,j). The distance between node k and the halfway point between node i and node j is $d(\mathrm{k}, \mathrm{m}_{\mathrm{i,j}})$.

Routing algorithms are employed to determine the next hop of $N_i$, $N_i^{+1}$. The distance between $N_i$ and its next hop $N_i^{+1}$ is *d*(i,+1). Likewise, the distance between $N_k$ and the halfway point between $N_i$ and $N_i^{+1}$ is $d(\mathrm{k}, \mathrm{m}_{\mathrm{i,+1}})$. A statement **power**(*d*(i,j)) in the pseudocode listing adjusts the transmit power according to the distance *d*(i,j). A statement **send** $(\mathrm{N_i} \longrightarrow \mathrm{N_j})$ indicates a packet forwarding from node i toward node j.

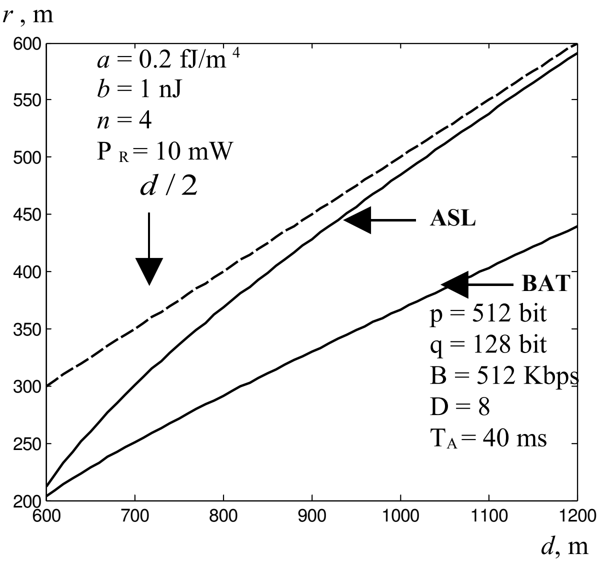


Fig. 4 Radius *r* scales with the distance for two MAC models

**Algorithm 1** $\mathrm{N_i^R} \leftarrow \mathrm{OneHop}(\mathrm{N_i})$

```
$N_i^R = \emptyset$
for $1 \le j \le n(N),\ j \ne i$ do
    if $d(i, j) \le R$
        $N_i^R = N_i^R \cup N_j$
    end if
end for
```

Algorithm 1 describes the procedure to determine the set $\mathrm{N_i^R}$, which includes the one-hop neighbours of $\mathrm{N_i}$. R denotes the communication range.

**Algorithm 2** $\mathrm{N_i^{+1}} \leftarrow \mathrm{NextHop}(\mathrm{N_i}, \mathrm{N_D}, \mathrm{N_i^R})$

```
if $N_D \in N_i^R$
    return $N_D$
end if
$s = (X^2 + Y^2)^{1/2}$
for $1 \le j \le n(N),\ j \ne i$ do
    if $N_j \in N_i^R$ and $d(j, D) < s$
        $N_i^{+1} = N_j,\ s = d(j, D)$
    end if
end for
```

Algorithm 2 applies the greedy routing algorithm to find the next relay of $N_i$.

The multihop service can be integrated into the routing algorithm.

**Algorithm 3** $\mathrm{MultiHop}\left(N_i, N_i^{+1}\right)$

```
do
    MULTI=0
    $d = d(i,+1)$
    $s = \left(X^2 + Y^2\right)^{1/2}$
    for $1 \le j \le n(N),\ j \ne i$ do
        if $d(j, m_{i,+1}) \le MIN(r,s)$
            $s = d(j, m_{i,+1})$ ,    $N_i^{+1} = N_j$ , MULTI=1
        end if
    end for
while MULTI
power $(d(i,+1))$
send $(N_i \to N_i^{+1})$
```

Algorithm 3 applies Theorem 1 or Theorem 2 to partition the communication link until suitable intermediate nodes are found. The procedure results in one forwarding.

**Algorithm 4** $\mathrm{Send}\ (N_S, N_D)$

```
$N_i = N_S$
do
    $\mathrm{NextHop}\left(N_i, N_D, N_i^R\right)$
    $\mathrm{MultiHop}\left(N_i, N_i^{+1}\right)$
while $N_i \ne N_D$
```

Algorithm 4 describes the successive approximation routing. The interaction between the routing procedure and the low-power forwarding is implemented via successive approximations. As soon as the routing algorithm determines the next hop, multihop optimization is applied to select an intermediate node. As soon as the packet is sent to the intermediate node, the routing algorithm is executed again. The multihop service algorithm itself is a successive approximation procedure as well.

In a two-hop distance approach, each node maintains a table of all immediate neighbors as well as each neighbor's neighbors. The number of hops taken into account determines the vulnerability of the routing in case of topology holes. However, considering more hops will require longer execution times. Fig. 5 shows how the transition from a single hop to two hops brings the execution time up. The code has been written in C and compiled for two CPUs: 8051 and Atmel AVR [25].

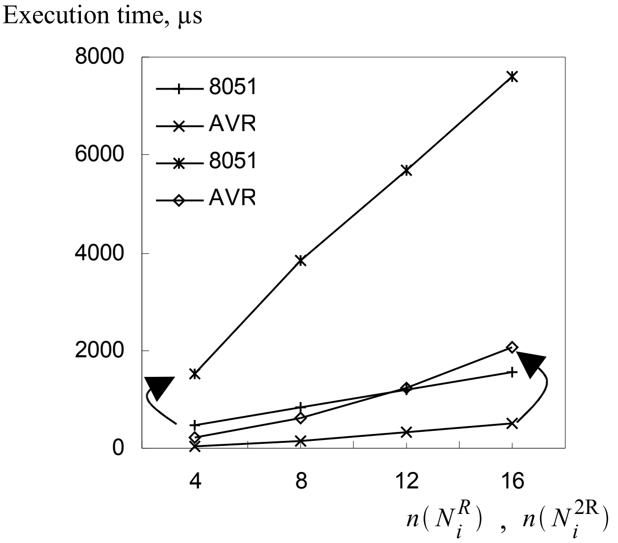


Fig. 5 Execution time to select the next relay

## VI. Security

The network functional partitioning into sensing, computation and communication can be used to deal with possible avenues of attacks. First, a misbehaving node may provide false sensor readings. In general, this kind of attack is not effective. Collected data is aggregated and a small number of malicious nodes can not change the profile of the physical event. However, a false alarm, an input has reached a threshold, will wake up several nodes and attack the batteries. Another attack related to the environment is a wrong location. Sensing is useful only in the context of where the data has been measured.

In contrast to sensing, a well placed enemy may successfully attack via wrong calculations. Aggregation is important for power efficiency and nodes that aggregate data packets are in a good position to attack.

Communication is what makes ad hoc networks most vulnerable and the multihop forwarding of packets unrolls ample possibilities for attackers. Once a malicious node has been included on the routing path, it will be in position to change the content of the packets. Along with data, packets may convey code. Mobile agent-based sensor networks distribute the computation into the participating leaf nodes [28], [29]. Since agents may visit a long path of nodes, a single modified packet can force several nodes to execute enemy code. Another axis along which packets can be affected relates to timing. A scheduling attack would change the number of past beacon periods a packet carries. Another form of a scheduling attack is delayed packets. An extreme type of this attack, termed black hole, is observed when a malicious node consumes packets. In a special case of black hole, an attacker could create a gray hole, in which it selectively drops some packets but not others. For example, the malicious node may forward control packets but not data packets.

## VII. REWARD Algorithm

REWARD (receive, watch, redirect) is a routing method that provides a scalable security service for geographic ad-hoc routing [33]-[35].

### *A.Black holes data base*

The algorithm creates a distributed data base for detected black hole and scheduling attacks. The data base keeps records for suspicious nodes and areas. The REWARD security service provides alternative paths for the geographic routing in an attempt to avoid misbehaving nodes and regions of detected attacks. The algorithm utilizes two types of broadcast messages, MISS and SAMBA, to recruit security servers. Security servers are nodes that keep records of the distributed data base and modify the geographic forwarding of packets to bypass insecure nodes and regions.

Assume that a demand-driven protocol performs a route discovery procedure. When the destination receives the query, it sends its location back and waits for a packet. If the packet does not arrive within a specified period of time, the destination node broadcasts a MISS (material for intersection of suspicious sets) message. The destination copies the list of all involved nodes from the query to the MISS message. Since the reason for not receiving the packet is most likely a black hole attack, all nodes listed in the MISS message are under suspicion. Nodes collect MISS messages and intersect them to detect misbehaving participants in the routes. The detected malicious nodes are excluded from the routing if other paths are available.

Radio is inherently a broadcast medium and nodes can detect black hole attacks if they listen to neighbor transmissions [32]. Fig. 6 shows an example. Each node tunes the transmit power to reach both immediate neighbors, $N_i^{+1}$ and $N_i^{-1}$ . We call this type of forwarding symmetrical. The nodes transmit packets and watch if the packets are forwarded properly. If a malicious node does not act as expected, the previous node in the path will broadcast a SAMBA (suspicious area, mark a black-hole attack) message.

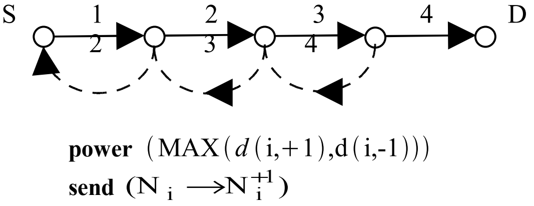


Fig. 6 Transmissions must be received by two nodes

Fig. 7 shows an example routing with the assumption that two malicious nodes would attempt a black hole attack. In this case the algorithm requires the nodes to listen for two retransmissions. Fig. 8 indicates the exact positions of two black holes in the path. The first malicious node forwards the packet using the required transmit power to deceive two nodes backward. The second malicious node drops the packet, however the attack is detected by the last node before the black holes. The missing transmission is shown by a dot line in Fig. 8. An extra black hole in the path would mask the attack.

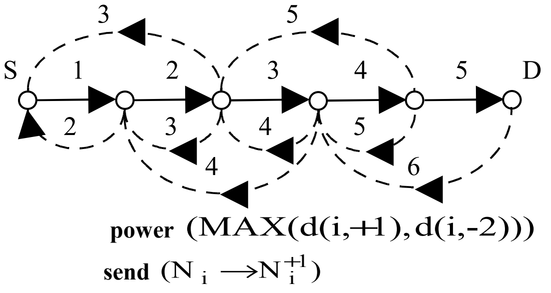


Fig. 7 REWARD against two black holes

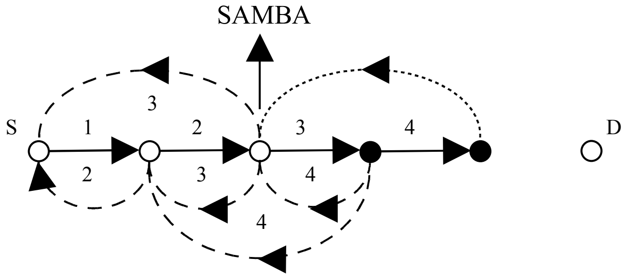


Fig. 8 REWARD detects the second black hole

In order to determine the effectiveness of REWARD we used ANTS (ad-hoc networks traffic simulator) [34],[35]. We assume that all nodes are stationary throughout the simulation. Fig. 9 shows simulation results of the throughput, 100 packets routing for eight example deployments. Each deployment has a density of 100 nodes randomly located in a square kilometer. The maximum communication range of the nodes is 100 meters. Also, the simulation results are obtained at 10% misbehaving nodes. MISS servers are recruited in a rectangular region. The source and destination locations define the diagonal of the rectangle.

Fig. 10 shows the fraction of malicious nodes detected against false detection. False detection is associated with nodes excluded from the network as malicious when in fact they are not. For the current simulation, nodes that are listed in two or more MISS messages are marked as malicious.

### *B. Energy overhead*

We distinguish between two types of security energy overhead. Static overhead is the additional energy required to watch for attacks. Dynamic overhead is the extra amount of energy spent to detect compromised nodes and mitigate routing misbehaviour. While the dynamic overhead will vary from application to application, the static overhead is a constant and an inevitable item in the energy budget.

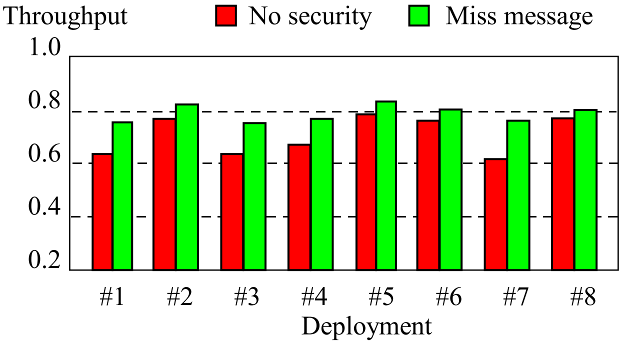


Fig. 9 The fraction of packets received for eight examples

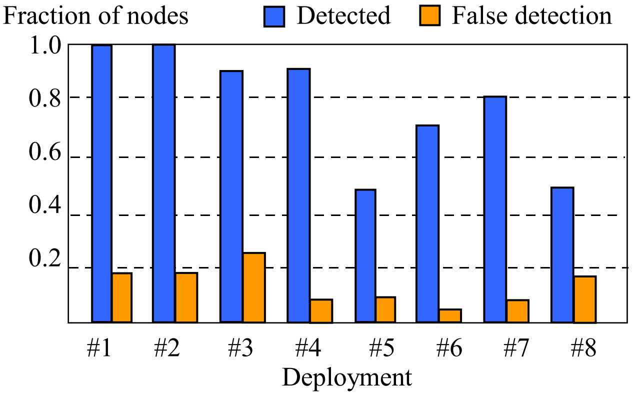


Fig. 10 Detected malicious nodes against false detection

Since secure routing protocols such as REWARD require symmetrical forwarding, the power efficiency is declined. Fig. 11 shows symmetrical routing for an example deployment. Three cases must be considered according to the distances:

$$d(i,-1)\le(d(i,+1))/2-r \qquad (11)$$

There is no security overhead in this case.

$$(d(i,+1))/2-r<d(i,-1)\le(d(i,+1))/2+r \qquad (12)$$

Again, there is no single-hop security overhead. Opportunities for partitioning of the link remain if neighbors are located within the shaded area, Fig. 11.

$$d(i,-1)>(d(i,+1))/2+r \qquad (13)$$

Symmetrical routing may not increase the energy , however partitioning of the link is not power efficient in this case.

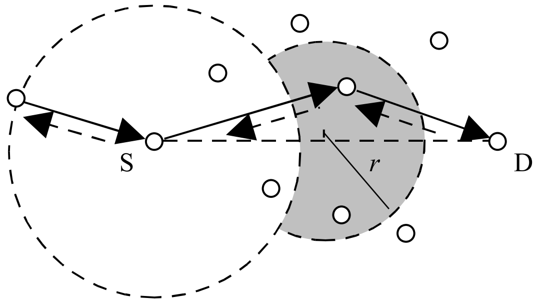


Fig. 11 Symmetrical routing

**Algorithm 5** MultiHopSym$(N_i, N_i^{+1})$

```
$s=(X^2+Y^2)^{1/2}$
if $d(i,-1)>(d(i,+1))/2+r$
  power $(MAX(d(i,+1), d(i,-1)))$
  send $(N_i \rightarrow N_i^{+1})$
  return
end if
if $d(i,-1)\le(d(i,+1))/2-r$
  for $1\le j\le n(N),\ j\ne i$ do
    if $d(j,m_{i,+1})\le MIN(r,s)$
      $s=d(j,m_{i,+1})$, $N_i^{+1}=N_j$
    end if
end for
  power $(d(i,+1))$
send $(N_i \rightarrow N_i^{+1})$
return
end if
for $1\le j\le n(N),\ j\ne i$ do
if $d(j,m_{i,+1})\le MIN(r,s)$ and
    $d(i,j)\ge d(i,-1)$
    $s=d(j,m_{i,+1})$, $N_i^{+1}=N_j$
end if
end for
power $(d(i,+1))$
send $(N_i \rightarrow N_i^{+1})$
```

Algorithm 5 provides multihop optimization for symmetrical routing.

## VIII. Conclusion

This paper manifests wireless ad hoc networks need multi-objective design. The multihop communication approach outlines tradeoffs between security, real-time and lifetime. We proposed a hierarchical communication model and employed it to compare how two MAC models are capable of link partitioning for non-regular topologies. The proofs can be used to organize look-up tables in the nodes memory and streamline the selection of the best next relay. We evaluated the static energy overhead associated with algorithms for secure routing, such as REWARD, which will help to reassess the lifetime of the network.

## References

[1] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", *ACM Wireless Networks J.*, vol. 8, no. 5, pp. 481-494, 2002.

[2] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks", *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493-506, June 2004.

[3] S. Biswas and R. Morris. "Opportunistic routing in multi-hop wireless networks", *ACM SIGCOMM Computer Communication Review*, vol. 34, Issue 1, pp. 69-74, Jan. 2004.

[4] D. Dewasurendra and A. Mishra, "Design challenges in energy-efficient medium access control for wireless sensor networks", in

*Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., CRC Press LLC, 2005, pp. 28-1–28-25.

[5] V. Zadorozhny, D. Sharma, P. Krishnmurthy, and A. Labrinidis, "Tuning query performance in mobile sensor databases," In *Proc. 6th Int. conference on Mobile data management*, Ayia Napa, Cyprus, pp. 247-251, 2005.

[6] Z. Karakehayov and N. Andersen, "Energy-efficient medium access for data intensive wireless sensor networks," In *Proc. Int. Workshop on Data Intensive Sensor Networks, IEEE ISBN: 1-4244-1241-2,* 8th Int. Conference on Mobile Data Management, Mannheim, Germany, pp. 116-120, May 2007.

[7] J. L. Gao, *Energy Efficient Routing for Wireless Sensor Networks*, Ph.D. dissertation, University of California, Los Angeles, 2000.

[8] J. M. Rabaey, M. J. Ammer, J. L. Silva, D. Patel and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking," *IEEE Computer*, vol. 33, pp. 42-48, Jul. 2000.

[9] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks* , 2, pp. 351-367, 2004.

[10] Z. Karakehayov, "Low-power communication for wireless sensor-actuator networks," In *Proc. of the Fifth IASTED Int. Conference on Communication Systems and Networks,* , Palma de Mallorca, Spain, ACTA Press, pp. 1-6, Aug. 2006.

[11] T. Moscibroda, R. O'Dell, M. Wattenhofer and R. Wattenhofer, "Virtual coordinates for ad hoc and sensor networks", ACM Joint Workshop on Foundations of Mobile Computing, Philadelphia, USA, October 2004.

[12] D. Puccinelli and M. Haenggi. "Wireless sensor networks: applications and challenges of ubiquitous sensing." *IEEE Circuits and Systems Magazine* , pp. 19-29, third quarter 2005.

[13] Zdravko Karakehayov, Knud Smed Christensen and Ole Winther, *Embedded Systems Design with 8051 Microcontrollers*, Dekker, 1999.

[14] T. Stoyanova, F. Kerasiotis, A. Prayati and G. Papadopoulos, "Evaluation of impact factors on RSS accuracy for localization and tracking applications", In *Proc. of the 5th ACM Int. Workshop on Mobility Management and Wireless Access*, Chania, Crete Island, Greece, pp. 9-16, Oct. 2007.

[15] V. Swaminathan, Y. Zou and K. Chakrabarty, "Techniques to reduce communication and computation energy in wireless sensor networks", in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., CRC Press LLC, 2005, pp. 29-1–29-34.

[16] M.T. Schmitz, B.M. Al-Hashimi and P. Eles, *System-Level Design Techniques for Energy-Efficient Embedded Systems*, Kluwer, 2004.

[17] Z. Karakehayov, "Dynamic clock scaling for energy-aware embedded systems," In *Proc. of the IEEE Fourth Int. Workshop on Intelligent Data Acquisition and Advanced Computing Systems*, Dortmund, Germany, pp. 96-99, Sept. 2007.

[18] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. Commun* ., vol. 32, no. 3, pp. 246-257, March 1984.

[19] I. Stojmenovic and Xu Lin, "Power aware localized routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122-1133, November 2001.

[20] Z. Karakehayov, "Low-power design for Smart Dust networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., CRC Press LLC, 2005, pp. 37-1–37-12.

[21] F. Baccelli, B. Blaszczyszyn and P. Muhlethaler, "An Aloha protocol for multihop mobile wireless networks", In *Proc. Of ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Antwerp. Belgium, 2004.

[22] Z. Karakehayov, "Security—lifetime tradeoffs for wireless sensor networks", In *Proc. 12th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Patras, Greece, pp. 646-650, Sept. 2007.

[23] E. M. Royer and C. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, pp. 46-55, April 1999.

[24] G. Zhou, T. He, S. Krishnamurthy and J. Stankovic, "Models and Solutions for Radio Irregularity in Wireless Sensor Networks," *ACM Transactions on Sensor Networks* , vol. 2, no. 2, pp. 221-262, 2006.

[25] Z. Karakehayov and Z. Monov, "Target-aware timing modelling for wireless ad-hoc networks," In *Proc. Int. Scientific Conference Computer Science'2006*, Istanbul, pp. 54-59, Oct. 2006.

[26] N. Sastry, U. Shankar and D. Wagner. "Secure verification of location claims." In *Proc. of the 2003 ACM Workshop on Wireless Security*, San Diego, September 2003.

[27] C. Karlof and D. Wagner. "Secure routing in wireless sensor networks: Attacks and countermeasures." In *Proc. of The First IEEE Int. Workshop on Sensor Networks, Protocols and Applications* , pp. 113-127, May 2003.

[28] H. Qi, S.S. Iyengar and K. Chakrabarty, "Multiresolution data integration using mobile agents in distributed sensor networks," *IEEE Trans. Syst., Man, Cybernetics Part C: Applic. Rev.,* 31(3), pp. 383-391, August 2001.

[29] Q. Wu, N.S.V. Rao, R.R. Brooks, S.S. Iyengar and M. Zhu, "Computational and networking problems in distributed sensor networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., CRC Press LLC, 2005, pp. 25-1–25-17.

[30] D. D. Hwang, B. C. Lai and I. Verbauwhede, "Energy-memory-security tradeoffs in distributed sensor networks", In *Ad-hoc, mobile, and wireless networks, Lecture Notes in Computer Science*, Eds., I. Nikolaidis, M. Barbeau and E. Kranakis, Springer, pp. 70-81, July 2004.

[31] Z. Karakehayov, "Design of distributed sensor networks for security and defense", In *Proc. of the NATO Advanced Research Workshop on Cyberspace Security and Defense: Research Issues*, (Gdańsk, September 6-9, 2004), J. S. Kowalik, J. Gorski and A. Sachenko, Eds., Springer, NATO Science Series II, v ol. 196, 2005, pp. 177-192.

[32] S. Marti, T. J. Giuli, K. Lai and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," In *Proc. 6th Int. Conference Mobile Computing Networking (MOBICOM-00)*, New York, ACM Press, pp. 255-265, August, 2000.

[33] Z. Karakehayov, "Using REWARD to detect team black-hole attacks in wireless sensor networks," in *Proc. Workshop on Real-World Wireless Sensor Networks*, REALWSN'5, Stockholm, June 2005.

[34] Z. Karakehayov and I. Radev, "REWARD: A routing method for ad-hoc networks with adjustable security capability," In *Proc. NATO Advanced Research Workshop "Security and Embedded Systems"*, Patras, pp. 180-187, August 2005.

[35] Z. Karakehayov and I. Radev, "A scalable security service for geographic ad-hoc routing," *Int. Scientific J. of Computing*, vol. 4, Issue 2, pp. 124-132, 2005.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 869–876

ISBN 978-83-60810-14-9
ISSN 1896-7094

# A Method of Mobile Base Station Placement for High Altitude Platform based Network with Geographical Clustering of Mobile Ground Nodes

Ha Yoon Song
Department of Computer Engineering, Hongik University, Seoul, Korea
Email: hayoon@wow.hongik.ac.kr

***Abstract*—High Altitude Platforms (HAPs) such as Unmanned Aerial Vehicles (UAVs) which can be deployed as stratospheric infrastructures enable a sort of new configurations of wireless networks. Ground nodes must be clustered in multiple sets and one dedicated UAV is assigned to each set and act as an MBS. For the intra-set nodes, UAVs must communicate each other in order to establish network links among intra-set nodes. Here we find a geographical clustering problem of networking nodes and a placement problem of MBSs. The clustering technique of mobile ground nodes can identify the geographical location of MBSs as well as the coverage of MBSs. In this paper we proposed a clustering mechanism to build such a configuration and the effectiveness of this solution is demonstrated by simulation. For a selected region with a relatively big island, we modeled mobile ground nodes and showed the result of dynamic placement of MBSs by our clustering algorithm. The final results will be shown graphically with the mobility of ground nodes as well as the placement of MBSs.**

## I. Introduction

THE recent development of Unmanned Aerial Vehicles (UAV) leads to interests in High Altitude Platforms (HAPs) as Mobile Base Stations (MBSs) of wireless wide-area networks. UAVs usually carry wireless network equipments and require less management and thus have been regarded as one of the best method to deploy wireless network over wide area without typical ground network equipments. This new sort of network configuration requires a lot of unsolved research topics from the physical layer to transportation layer as well as the ideal configuration of network. Many researches have been concentrated on the communication between stratospheric UAVs and mobile ground nodes. These topics include establishments of communication links among UAVs as well.

The goal of HAP based network is to cover as wide area as possible with deployment of multiple UAVs, i.e. ultimate goal of HAP network is to deploy as many MBSs to cover dedicated area in order to construct a network structure. In this configuration, UAVs can act as mobile base stations for the network. This sort of network configuration raises a new configuration problem of UAVs and mobile ground nodes.

Here we suggest an idea. Mobile ground nodes consist a number of clusters in order to be served by MBSs and each MBS covers a dedicated cluster of mobile ground nodes. MBSs cooperate each other in order to support communication of mobile ground nodes in the whole area. The HAP based wireless network usually regarded as a viable solution for the networks with minimal ground infrastructures or countries under development and so on.

In this paper, we will show a method to deploy MBSs for a dedicated area with possible number of MBSs and to cover dedicated area efficiently. By adopting a clustering mechanism for mobile ground nodes, we can solve a placement problem of MBSs as well as the coverage of each MBS. For an island area we set mobility model and simulated the dynamic clustering of mobile ground nodes and find proper locations for MBSs and coverage.

This paper is structured as follows. In section II we will discuss research basis for this paper. In section III we will see the basic configuration concept of HAP based network assumed in this paper. In the following section IV we will explain our clustering algorithm for mobile ground nodes and MBS placement. Then we will show simulation experiment and results in section V. The final section will conclude this paper and will discuss the future research direction.

## II. Related Works

The basis of this research is usually categorized into two basic parts. The first one is HAP based networks. The second one is clustering algorithms. We will discuss about this two topics simply.

### A. HAP based Network

HAP based network is one of the most recent research topics while there have been a long idea about stratospheric platforms as a media of networking. With the development of network technologies as well as aerial industries, HAP can be actual one and the idea of HAP based network spawns into real world. Nowadays, most of countries research on this topic. Some of them regard this sort of network as national project while others act as an individual company bases. The Republic of Korea researches on this topic as a national project as shown in [1] with the fruitful research at ETRI (Electronics and Telecommunication Research Institute).

This work was supported by 2008 Hongik University Research Fund.

Japan has the similar situation. Japanese Aerospace Exploration Agency [2] has an ongoing effort with HAP based network, named as SkyNet project.

For EU nations or EU companies, there are projects such as HeliNet and CAPANINA. England was one of the early starter in this topic and most of EU nations are participated in those projects [3]. The Australian continent also has academic research on this topic [8].

Also in the US, projects such as Sky Station, High Altitude Long Operation, SkyTower, Staratellite, Weather Balloon HAPs are undergoing ones [4].

The characteristics of this project are that they are still assuming a single HAP platform and are mostly concentrated on communication links establishment.

For example, WRC started the first specification of frequency allocation for HAP network since the year 1997 and several bandwidth frequencies are allotted for specific nations [9]. By ITU, following radio frequencies and specifications for the Republic of Korea are allotted.

- Altitude 20.6 - 23.8 Km
- Frequency 47.9 - 48.2 GHz and 47.2 - 47.5 GHz
- Bandwidth 500Mhz

There have been several researches on the communication link establishment between mobile ground nodes and HAP base station [5], [6]. Also a protocol standard such as 802.16 can be a possible candidate for HAP networks [7]. For the inter-HAP links, there also have been researches such as [8].

Apart from these researches, this paper assumes an environment of multiple heterogeneous HAPs. We regard each HAP as a MBS of mobile ground nodes and tried to cluster mobile ground nodes in order to solve the placement problem of MBS. As far as we know, no such topic has been dealt until the first submission of this paper. The most similar one is in [10] but it is on a positioning control for HAP station.

### B. Clustering Algorithms

In a field of data mining, there have been researches regarding clustering of data in a large Database [11]. And multiprocessor versions can be found in [12]. They are usually concentrated on the patterns of related data while we need a geographical clustering based on the ground node coordinates. Thus we screened out several candidates and tried combinations of those clustering algorithms.

The most prominent candidates are K-mean, BIRCH, EM, PROCLUS, and SUBCLUS. The K-mean is one of the most popular algorithm in the clustering world. Since we are dealing with mobile ground nodes, we cannot identify the exact coordinates but we only assume the probability of node coordination. In this aspect, we concentrated on Expectation Maximization with a probability of node coordinates.

Considering the limitations of number of nodes in a cluster, that is actually a bandwidth limitation of wireless routers equipped on HAP MBS, We need to split a cluster or merge clusters in order to balance the number of nodes in a cluster. The BIRCH and their successors can cope with such a requirement, thus we choose it as a possible candidate.

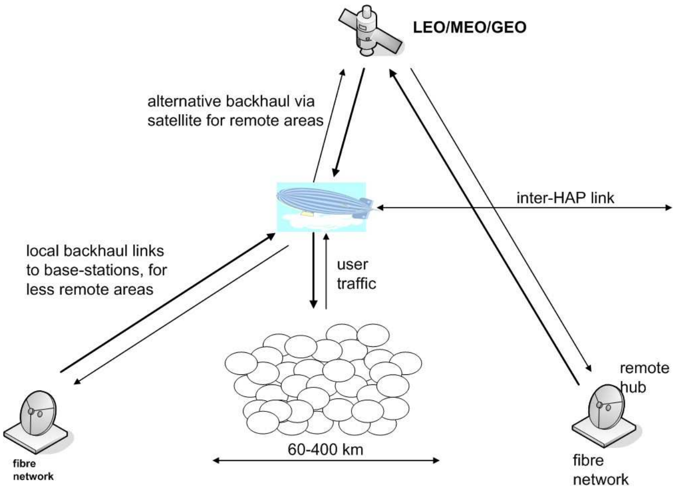

LEO/MEO/GEO
alternative backhaul via satellite for remote areas
inter-HAP link
local backhaul links to base-stations, for less remote areas
user traffic
remote hub
60-400 km
fibre network
fibre network

Fig. 1. Basic Configuration of HAP based Network

Even though EM and BIRCH are nice candidates for our application, these two cannot deal ideal initial clustering and the results are usually not a geographical one. Thus we need a multiphase clustering algorithm. Usually initial clusters can be drawn by K-mean algorithm and then the core cluster algorithm such as BIRCH or EM can work. For geographical consistency, a post processing of merging or splitting must be done. Two algorithms of SUBCLUS and PROCLUS are typical examples of multiphase clustering algorithms that attracts our interest.

However, apart from this existing algorithms, we will study the most appropriate clustering algorithms for HAP MBS placement. In this paper, we will show a basic result based on K-mean algorithm [13].

## III. Basic Concept of HAP based Network Configuration

The concept of HAP networking is similar to that of satellite communication. Characteristics such as Broadness, Simultaneousness, Flexible Configuration, and Broadbandwidthness of HAP based network are similar to that of satellite network, however, on-time supplements, ease of management, short communication distance, low power mini handset, low round trip time are typical characteristics of HAP network that cannot be found in satellite network. The key point here is low communication delay and flexible network configuration. In this paper, we spotted on flexible network configuration and will start the discussion from the basic configuration of HAP network. Figure 1 shows a basic configuration of HAP based network. HAP usually resides on stratospheric area, about 21km altitude with very low speed of winds as reported by NASA, USA.

Even though figure 1 shows only one HAP device, however there could be a lot of plans for multiple HAPs. This configuration with multiple HAPs will cause a interplatform link study and reapplication of traditional frequency reuse problem. It has been assumed that a Radio or a Optical communication

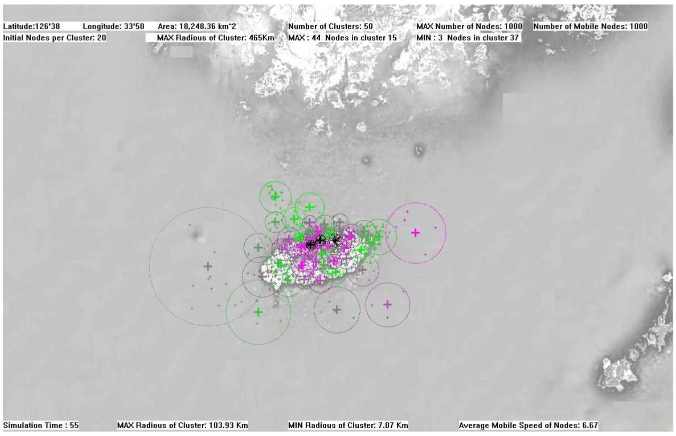

Latitude:126'38
Longitude: 33'50
Area: 18,248.36 km^2
Number of Clusters: 50
MAX Number of Nodes: 1000
Number of Mobile Nodes: 1000
Initial Nodes per Cluster: 20
MAX Radious of Cluster: 465Km
MAX : 44 Nodes in cluster 15
MIN : 3 Nodes in cluster 37
Simulation Time : 55
MAX Radious of Cluster: 103.93 Km
MIN Radious of Cluster: 7.07 Km
Average Mobile Speed of Nodes: 6.67

Fig. 2. Geographical Simulation Results mapped on Cheju Island and Neighboring Seas

for interplatform link and it is another research area. Also a routing problem can be arisen while there are a lot of prepared routing protocol such as OLSR, TBRPF, DSDV, CGSR, WR, AODV, DSR, LMR, TORA, ABR, SSR, and ZRP.

Therefore we will concentrate on the multiple HAPs network. The first problem of this multiple HAPs network is a clustering of mobile ground nodes and finds an optimal location for multiple HAPs as a clusterhead or mobile MBS. We expect the final result looks like in figure 2 where we can see 50 MBSs work as clusterhead and the coverage of each MBS are defined by the population density.

For the environment of this multiple HAPs network, we have similar problems in wireless network listed below.

- Uncertainty of mobile ground node coordinates
- Link Error Rate, lower than satellite network
- Network Topology Variance, frequent
- Network Security

We can consider parameters below for network configuration.

- Total number of mobile ground node, network size N
- Network connectivity
- Topological rate of change
- Link capacity
- Fraction of unidirectional links
- Mobility of ground nodes
- Fraction and frequency of sleeping nodes

## IV. Clustering of Ground Nodes for MBS Placement

For multiple HAP network, mobile ground nodes are clustered into several clusters, and each HAP acts as MBS of a cluster or is regarded as clusterhead. Since the number of nodes in a cluster is a major parameter, another parameter of cluster coverage is a dependent variable. Each HAP has capability to adjust its coverage individually by adjusting its angle of elevation. These parameters can vary dynamically according to the mobility of ground nodes. We can assume the following scenarios of dynamic clustering.

- Initial clustering: The very first stage of network configuration. Almost a static clustering.
- Service oriented reclustering: New influx of mobile ground nodes to a specific cluster cause a overloaded router on a HAP and thus requires a reclustering. Maybe a new UAV can be added.
- Geographical reclustering: The mobility of ground nodes can cause more area to be covered by HAP based network. This situation causes reclustering.
- Airship backup reclustering: Failure of one or more UAV can cause a reclustering situation.
- Shrinking clustering: Decrement of the number of mobile ground nodes will cause a non-mandatory reclustering.

From the scenarios above, we assume a geographical reclustering situation in this paper. However all of the above scenarios require reclustering algorithm according to the geographical network coverage or mobile ground node distribution. In order to provide such an algorithm, we can assume the following requirements for network parameters.

1) An UAV as a HAP can equip MBS facilities.
2) An UAV can identify number of ground nodes served.
3) An UAV can identify the location of mobile ground nodes.
4) An UAV can identify the location of itself.
5) An UAV can vary the geographical network coverage by adjusting the angle of elevation.
6) An UAV can communicate with ground base stations.

These assumptions can be realized with the help of aero engineering, electro engineering, antenna technology or other related modern technology. Under these assumptions, the following lists requirements for clustering algorithms of mobile ground nodes.

1) The sum of bandwidth requirement from mobile ground nodes in a cluster is equal to or smaller than the total bandwidth of an MBS for the cluster.
2) Each UAV can move in a limited speed. Thus the reclustering must consider the speed of UAVs.
3) Cluster algorithm requires realtimeness for continuous network service.

Here we can show a simple clustering algorithm based on K-mean clustering algorithm.

The exact location of each MBS for a cluster is a centroid of the each cluster. The centroid of a cluster is median value of geographical coordinates of mobile ground nodes in the cluster. Instead of mean value, we use median value in order to guarantee the efficiency of MBS coverage. Thus we can use this algorithm for MBS placement.

Even though we use K-mean algorithm, this algorithm can be replaced any of proper clustering algorithm. We experienced several clustering algorithms produce inclusive clusters. To resolve such problem, we need postprocessing of clusters that merges inclusive clusters to including cluster. We regard this problem as a future topic and will discuss basically in the conclusions and future research section.

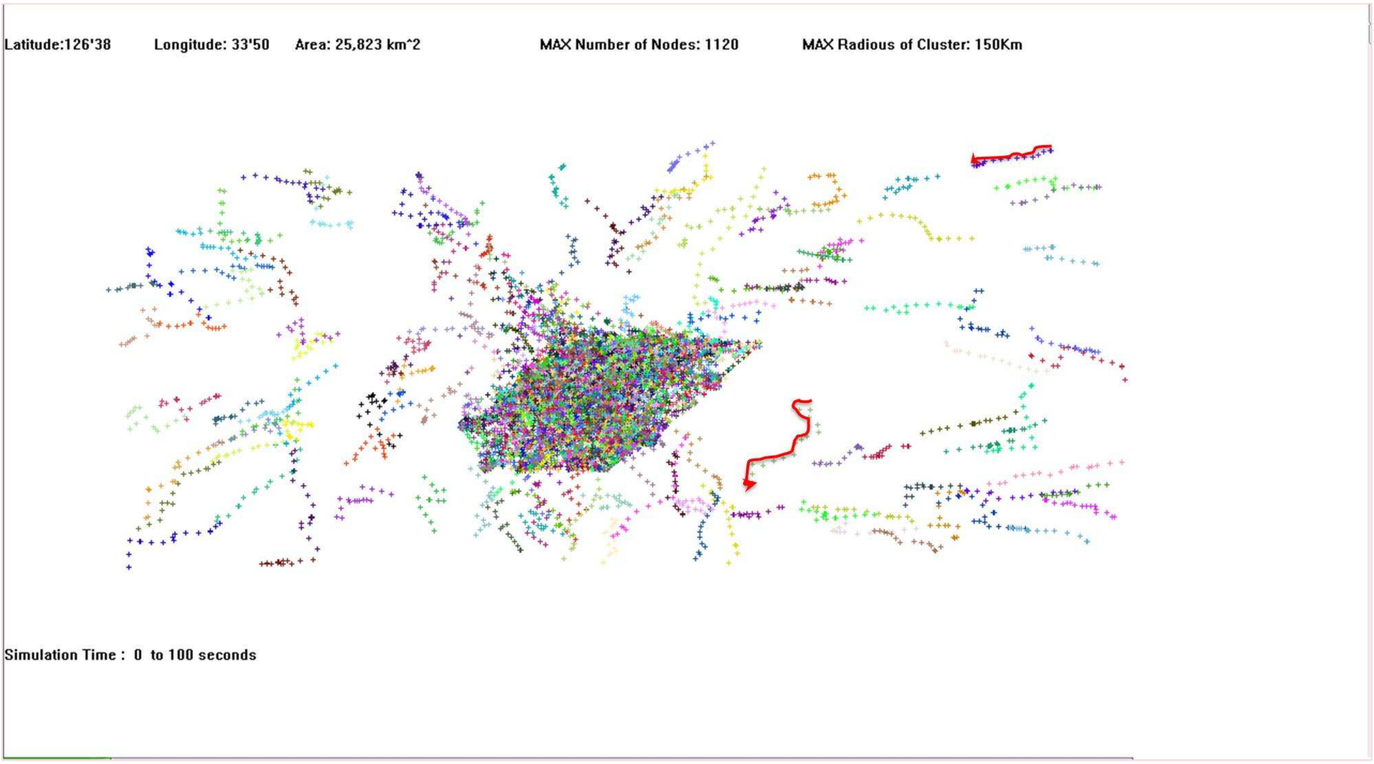

Latitude:126'38
Longitude: 33'50
Area: 25,823 km^2
MAX Number of Nodes: 1120
MAX Radious of Cluster: 150Km
Simulation Time : 0 to 100 seconds

Fig. 3. Node Mobility Snapshot

**Algorithm 1** Clustering and MBS Placement

```
Require: B: Router Bandwidth Capacity on a HAP
    S: Total sum of required bandwidth by nodes in a cluster
    C: A set of coordinates of mobile ground nodes
    A: Total area size to be covered by network
    M: Maximum area can be covered by a HAP

Ensure: L: List of clusters
    P: List of cluster centroids

    SpB = S/B //Bandwidth requirement

    ApM = A/M //Geographical requirement

    Calculate the number of Cluster K = min(SpB,ApM)

    L = result of K-mean clustering algorithm with input C

    P = calculated centroid of each cluster.
```

The time complexity of algorithm 1 is $O(TN^2)$ where N is total number of mobile ground nodes and T is number of reclustering.

## V. Simulation of HAP MBS Placement

We do simulation experiment based on assumptions and algorithms for HAP based network. Our aim is to show simulation results geographically in order to identify clustering result and HAP placement result graphically. The simulation is done first by NS-2 with mobility models for nodes. With the NS-2 trail output, we parse them in order to identify node mobility in time and the parsed results are used by clustering algorithms. Finally a clustering results are visualized.

### *A. Simulation Environments and Parameters*

For the geographical environment for network simulation, we choose Cheju Island of the Republic of Korea and its neighboring seas. 1000 nodes are distributed according to population distribution on Cheju Island and several marine vehicles were also assumed. The reason why we choose this area is as follows.

- Two cities have denser population than other area.
- The other area has lower population.
- There is high mobility to or from two cities to other area.
- Almost all sort of mobility can be included. There are various type of mobility including human walk, ground vehicle, horse rider, and marine vehicle.
- There are frequent mobility between Cheju island and its two neighboring small islands which have lower population.
- Lots of travelers show very high mobility.

In addition we wanted to look at the phenomena that non-island areas are connected from island area by HAP MBS. The simulation parameters are as follows:

- 28,246 $Km^2$ coverage area
- Total number of mobile ground node is 1000
- Number of initial clusters is 50

- Number of average nodes in a cluster is 20
- Random Waypoint (RW) mobility model
- Speed of mobility is 4∼7 km/sec
- Node population is proportional to actual population density
- Simulation time up to 60 minutes
- Maximum 465Km for inter-hap links
- Maximum 150Km for cluster radius as specified by ITU [14]

Figure 3 shows a cumulated snapshot of mobile nodes trajectory for these simulation experiments.

### B. Simulation Results and Analysis

With this simulation parameters, simulation results are shown geographically in selected figures 4, 5, 6, 7, 8.

Each figures stands for the result of simulation from 15 minutes to 55 minutes after the start of simulation. In each figure, the radius of a cluster is magnified by the factor of 1.4 for visibility reason. The dots are mobile ground nodes while crosses are centroids of clusters and thus locations of HAP MBSs. Two cities with high population show a lot of clusters with small coverage. This is due to the restriction of network bandwidth provided by on HAP MBS. In order to guarantee minimum bandwidth for individual nodes, the maximum number of nodes per cluster is restricted since HAP MBS has limited bandwidth as a wireless router. The neighboring seas are covered by relatively larger clusters and smaller number of HAP MBS. The maximum number of nodes in a cluster is up to 47 while the minimum number is 2 for a cluster. The largest cluster has radius of 154Km in marine area while the smallest cluster has 6.5Km in city area. Note that 154Km of cluster radius slightly exceeds a maximum radius of HAP MBS coverage specified by ITU.

Even though some nodes look like multiple participant in a cluster, they are actually a member of specific single clusters. This sort of overlapping can be overcome by frequency allocation and reuse. Some clusters with dedicated MBSs look like an orphan. However, the presumably orphan MBSs are within 930Km of other connected HAP MBS, i.e. orphans are within the range of inter-hap links.

Some postprocessing algorithms required in order to merge or to split clusters with abnormal number of nodes. Clusters with larger number of nodes must be split and clusters with smaller number of nodes must be merged. The ultimate goal of post processing is to assign adequate number of nodes equally likely to clusters. We experienced the overloaded clusterhead (MBS) is usually a problem for network configuration. We will provide a simple postprocessing algorithm in section VI

We experienced less than 1 second for each clustering with usual Pentium based personal computers. We believe this guarantees realtimeness of our clustering algorithm.

## VI. Conclusion and Future Research

In this paper, we present a HAP MBS placement solution over multiple HAP based wireless network. Considering the number of ground nodes and area coverage, mobile ground nodes are clustered in order to prepare placement location. As a result, each HAP based MBS can be placed at geographically suitable location in order to serve as a BS of a cluster. We experiment these scenarios by simulation and the results are visualized in a specific area of Cheju Island.

However, we can find several problems regarding the clustering results. First, we must suppress of mobility of MBSs for stable network service. Even if we assumed mobility of HAP MBSs, high mobility of existing MBSs would cause instable network service.

Second, other than K-mean algorithm, more sophisticated algorithm must be introduced. We are now focusing on Expectation Maximization (EM) and BIRCH [11], [12] as a core algorithm for HAP MBS placement. For uncertain coordinates of highly mobile nodes we can adjust probabilities for such nodes to be members of a specific cluster with the speed of mobile node or so. Then this probability set will be directly applied to EM based clustering.

Third, the number of nodes per cluster must be distributed evenly for stable network bandwidth allocation. Since there are no clustering algorithms that calibrate the number of elements per cluster, a new clustering algorithm must be introduced. And we sometimes observe inclusive clusters and semi-inclusive (highly overlapped) clusters in simulation results. BIRCH has the abilities of splitting or merging clusters by use of CF tree. These capabilities can be a useful method to assign evenly distributed number of nodes to clusters.

These combination of clustering algorithms ultimately leads to a multiphase clustering algorithm. We are now considering the following process.

1) Preprocessing with static clustering algorithms such as K-mean.
2) Main clustering with dynamic clustering algorithms such as EM.
3) Postprocessing for split and merge of clusters with algorithms such as BIRCH.

We believe the outmost clustering algorithm would be BIRCH since it requires another clustering algorithm for core clustering features.

For postprocessing as mentioned in section IV, we have the following idea. We can merge a cluster into outer cluster when the cluster is inclusive under the condition that $|C_1 - C_3| + R_3 < R_1$ and $N_1 + N_3 < N$, where $C_i$ is a coordinate for cluster $i$, $R_i$ is a radius for cluster $i$, and $N_i$ is number of nodes in a cluster $i$. The first term stands for Cluster $C_3$ is totally inclusive in Cluster $C_1$ and the second term restricts merge if the total number of nodes in two clusters exceeds a desired number of nodes $N$ in a cluster.

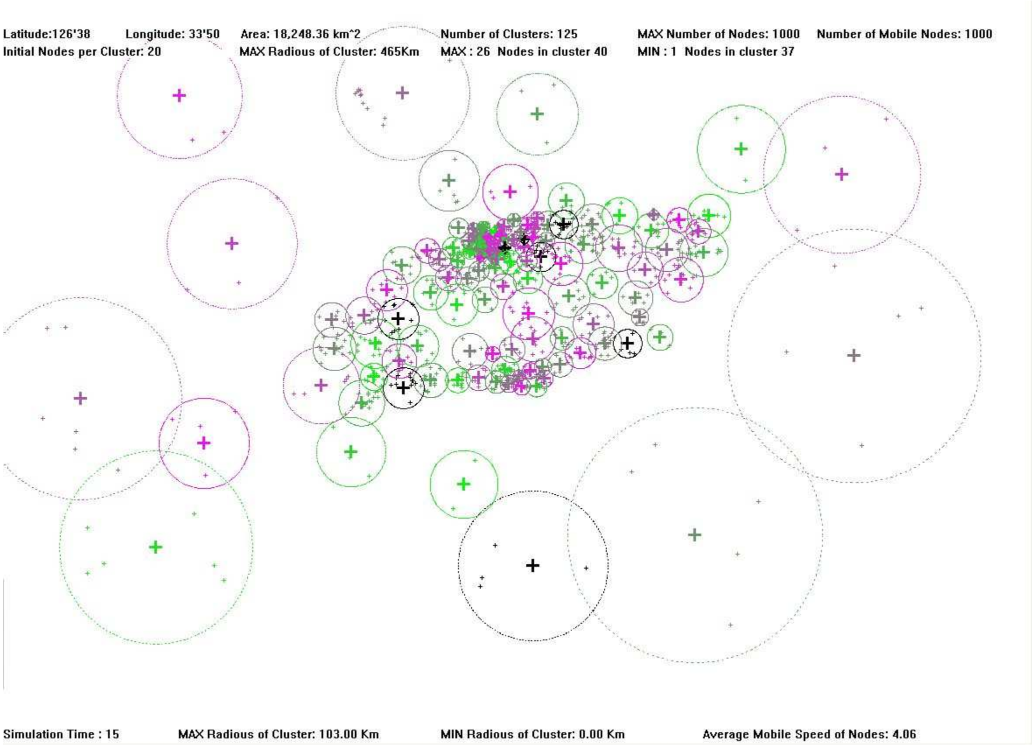
Similarly we can merge semi-inclusive clusters where most part of a cluster, e.g. 75%, is covered by another cluster.

## References

[1] Korea Radio Promotion Association, http://www.rapa.or.kr.

[2] Japan Aerospace Exploration Agency, http://www.jaxa.jp.

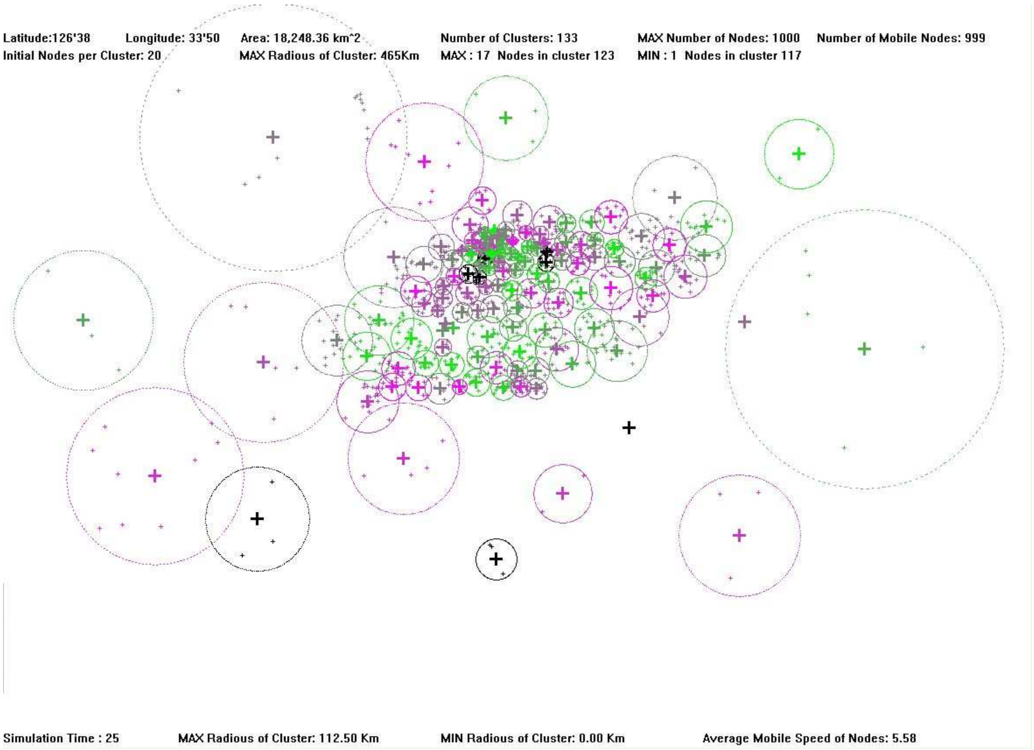
[3] J. Thornton, D. Grace, C. Spillard, T. Konefal and T. C. Tozer, "Broadband Communications from a High-altitude Platform," *Electronics and Communication Engineering Journal,* Vol. 13, No. 3, pp. 138–144, 2001.



```
Latitude:126'38    Longitude: 33'50    Area: 18,248.36 km^2          Number of Clusters: 125         MAX Number of Nodes: 1000    Number of Mobile Nodes: 1000
Initial Nodes per Cluster: 20          MAX Radious of Cluster: 465Km  MAX : 26  Nodes in cluster 40   MIN : 1  Nodes in cluster 37

Simulation Time : 15    MAX Radious of Cluster: 103.00 Km    MIN Radious of Cluster: 0.00 Km    Average Mobile Speed of Nodes: 4.06
```

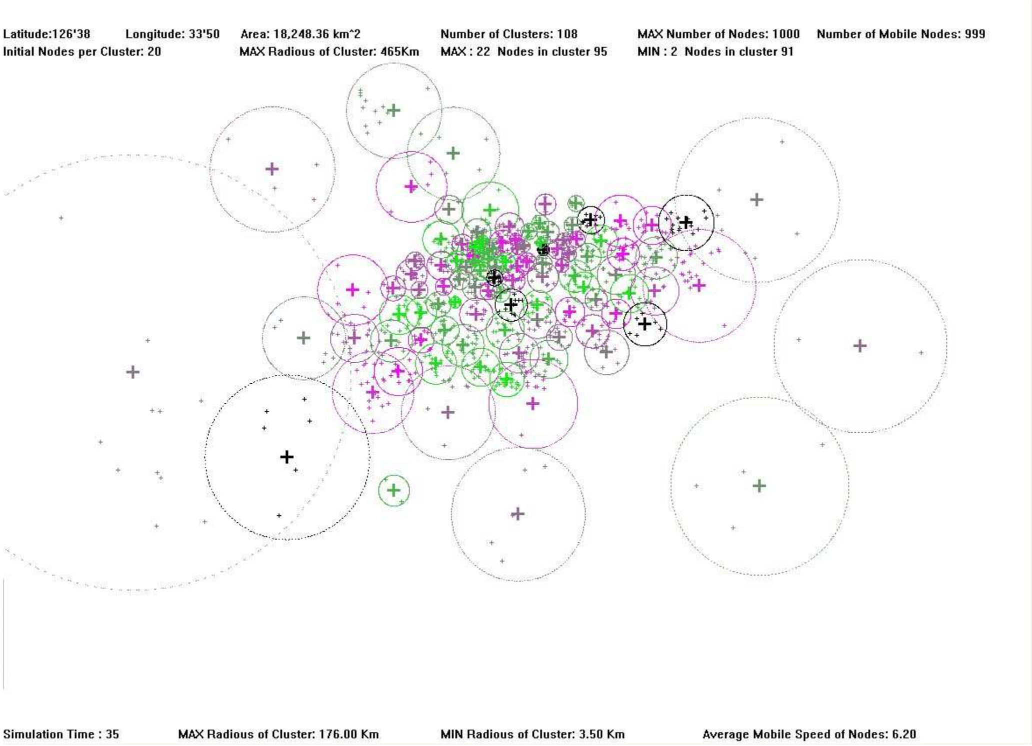
Fig. 4. Placement and Coverage of MBSs after 15 minutes



```
Latitude:126'38    Longitude: 33'50    Area: 18,248.36 km^2          Number of Clusters: 133          MAX Number of Nodes: 1000    Number of Mobile Nodes: 999
Initial Nodes per Cluster: 20          MAX Radious of Cluster: 465Km  MAX : 17  Nodes in cluster 123  MIN : 1  Nodes in cluster 117

Simulation Time : 25    MAX Radious of Cluster: 112.50 Km    MIN Radious of Cluster: 0.00 Km    Average Mobile Speed of Nodes: 5.58
```

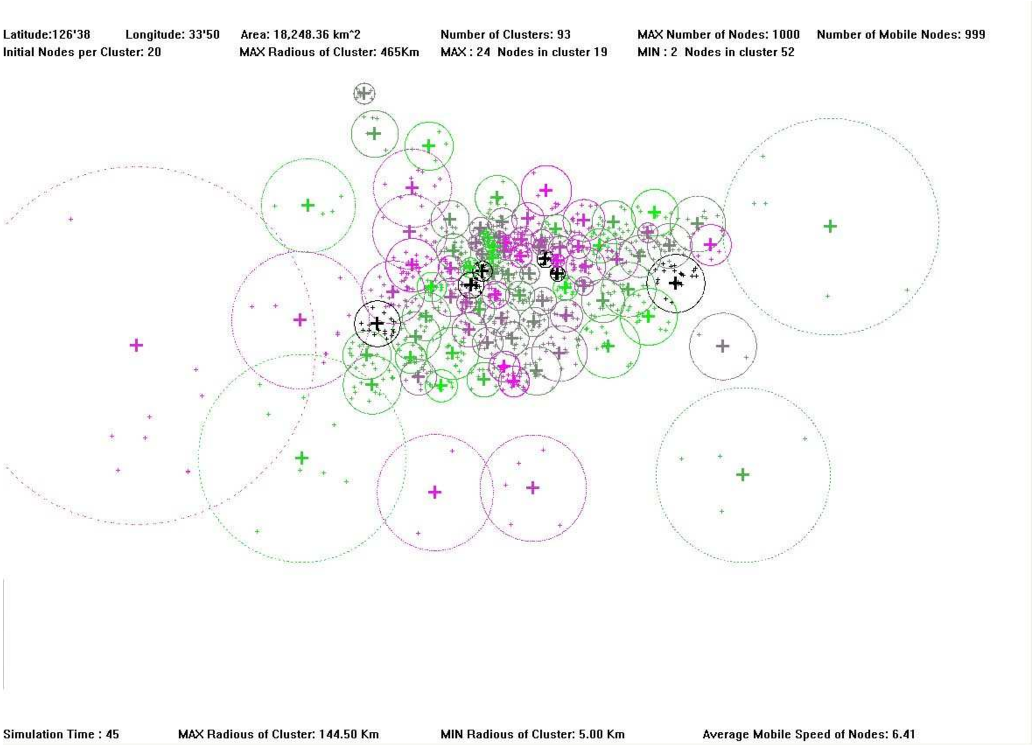
Fig. 5. Placement and Coverage of MBSs after 25 minutes



Latitude:126'38 Longitude: 33'50 Area: 18,248.36 km^2 Number of Clusters: 108 MAX Number of Nodes: 1000 Number of Mobile Nodes: 999
Initial Nodes per Cluster: 20 MAX Radious of Cluster: 465Km MAX : 22 Nodes in cluster 95 MIN : 2 Nodes in cluster 91

Simulation Time : 35 MAX Radious of Cluster: 176.00 Km MIN Radious of Cluster: 3.50 Km Average Mobile Speed of Nodes: 6.20

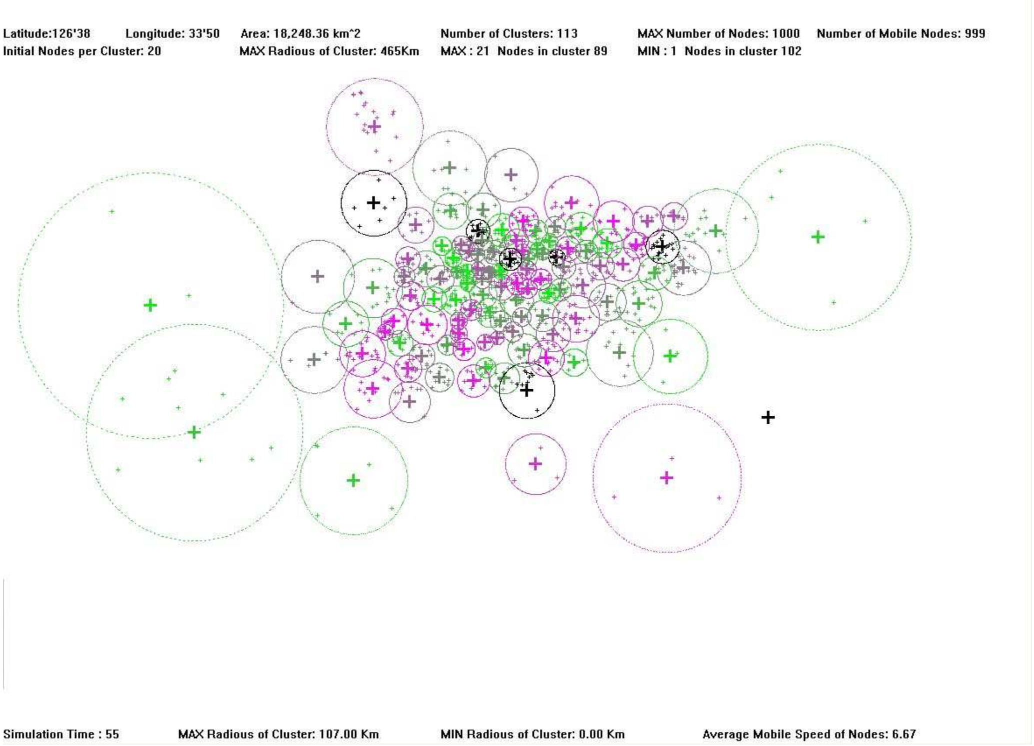
Fig. 6. Placement and Coverage of MBSs after 35 minutes



Latitude:126'38 Longitude: 33'50 Area: 18,248.36 km^2 Number of Clusters: 93 MAX Number of Nodes: 1000 Number of Mobile Nodes: 999
Initial Nodes per Cluster: 20 MAX Radious of Cluster: 465Km MAX : 24 Nodes in cluster 19 MIN : 2 Nodes in cluster 52

Simulation Time : 45 MAX Radious of Cluster: 144.50 Km MIN Radious of Cluster: 5.00 Km Average Mobile Speed of Nodes: 6.41

Fig. 7. Placement and Coverage of MBSs after 45 minutes



Latitude:126'38 Longitude: 33'50 Area: 18,248.36 km^2 Number of Clusters: 113 MAX Number of Nodes: 1000 Number of Mobile Nodes: 999

Initial Nodes per Cluster: 20 MAX Radious of Cluster: 465Km MAX : 21 Nodes in cluster 89 MIN : 1 Nodes in cluster 102

Simulation Time : 55 MAX Radious of Cluster: 107.00 Km MIN Radious of Cluster: 0.00 Km Average Mobile Speed of Nodes: 6.67

Fig. 8. Placement and Coverage of MBS after 55 minutes

[4] A. K. Widiawan and R. Tafazolli, “High Altitude Platform Station(HAPS): A Review of New Infrastructure Development for Future Wireless Communications,” Wireless Personal Communications, Vol. 42, pp. 387–404, 2007.

[5] D. Grace, C. Spillard, J. Thornton, T. C. Toze, “Channel Assignment Strategies for a High Altitude Platform Spot-beam Architecture,” The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 4, pp. 1586–1590, 2002.

[6] Durga Shankar Dasha, Arjan Durresi and Raj Jain, “Routing of VoIP traffic in multi-layered Satellite Networks,” *Performance and control of next-generation communications networks*, Vol. 5244, pp. 65–75, 2003.

[7] Floriano De Rango, Andrea Malfitano, Salvatore Marano, "PER Evaluation for IEEE 802.16-SC and 802.16e Protocol in HAP Architecture with User Mobility under Different Modulation Schemes," IEEE Globecom, pp. 1–6, 2006.

[8] Robert Andrew Janis Purvinskis, Interplatform Links, Univ. South Australia, Adelaide, Australia, 2003.

[9] Dudley Labą’s list of Frequency Allocations, http://www.dudleylab.com/freqaloc.html.

[10] Toshiaki Tsujii, Jinling Wang, Liwen Dai, Chris Rizos, “A Technique for Precise Positioning of High Altitude Platforms System (HAPS) Using a GPS Ground Reference Network,” *14th Int. Tech. Meeting of the Satellite Division of the U.S. Inst. of Navigation*, pp. 1017–1026, 2001.

[11] J. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations,” *Proc. Fifth Berkeley Symposium on Math. Statistics and Probability*, Vol. 1, pp. 281–297, 1967.

[12] I. S. Dhillon and D. S. Modha, “A Data-clustering Algorithm on Distributed Memory Multiprocessors,” *Large-Scale Parallel Data Mining*, Vol. 1759, pp. 245–260, 1999.

[13] Vance Faber, “Clustering and the Continuous K-means Algorithm,” Los Alamos Science, No. 22, 1994.

[14] Karapantazis S. Pavlidou F.-N, “The Role of High Altitude Platforms in beyond 3G Networks,” *IEEE Wireless Communications*, Vol. 12, No. 6, pp. 33–41, 2005.

Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 877 – 883

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Energy Efficient Percolation-Driven Flood Routing for Large-Scale Sensor Networks

Gergely Vakulya
University of Pannonia,
Veszprém, Hungary
Egyetem u. 10.
Email: vakulya@dcs.vein.hu

Gyula Simon
University of Pannonia,
Veszprém, Hungary
Egyetem u. 10.
Email: simon@dcs.vein.hu

*Abstract*—**Flooding algorithms are widely used in ad-hoc sensor networks, especially as building blocks of more sophisticated routing protocols. A distributed percolation-driven energy efficient probabilistic flood routing algorithm is proposed, which provides large message delivery ratio with small number of sent messages, using neighborhood information only. The performance of the algorithm is analyzed and the theoretical results are verified through simulation examples.**

## I. Introduction

Wireless sensor networks are made of large number of inexpensive nodes with limited processing and communication capabilities, and have limited energy reserves. Such networks have been used in a wide range of applications with various goals, using different platforms and technology [1][2][3][4]. Independently of the nature of the application, however, certain middleware services are always present, one of them being routing. Limited resources on the sensor nodes require that routing protocols be simple, energy-conserving and robust.

Depending on the actual deployment scenario and available hardware, several ad-hoc routing algorithms have been proposed. Location-aware routing schemes, e.g. greedy perimeter stateless routing [5], location aided routing [6], distance routing effect algorithm [7] use position information of the nodes obtained from GPS or other localization services. Other data centric schemes do not use location information, e.g. directed diffusion [8], ad-hoc on-demand distance vector routing (AODV) [9], dynamic source routing [10], temporarily ordered routing [11], or zone routing [12], just to name a few. To allow sensor networks better cope with scaling, hierarchical protocols were designed, e.g. Low-Energy Adaptive Clustering Hierarchy (LEACH) [13] and Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [14].

The simplest routing protocol is flooding [15], which is a useful simple (but inefficient) protocol in itself, and also is a building block for several more sophisticated algorithms, e.g. directed flood-routing [16], controlled flood routing [17], tiny ad hoc routing [18], or AODV [9], which is used in Zigbee's routing protocol [18].

The basic flooding protocol is the following:

- When a source node intends to send a message to all other nodes in the network, it broadcasts the message to its neighbors.
- When a node first receives a message it rebroadcasts the same message to its neighbors. All other copies of the same message received later will be discarded by the recipient node.

Flooding naturally can be used in its basic form for network-wide dissemination of information. In this context the important performance criteria of the algorithm are delivery ratio, latency, and the energy efficiency. Flooding is also used for route discovery [9]. Route discovery protocols find a route from a source node to a destination node in two phases. First the network is flooded by a discovery request message originated from the source, and then a reply message is sent back from the destination to the source node. In this way the network learns the path from the source to the destination and this route will be used in the subsequent communication sessions. In this context the quality – primarily the length – of the discovered route is an important performance criterion as well.

Apart from its simplicity the main advantage of the flood routing protocol lies in its robustness: the implicit redundancy built in the algorithm provides resistance against high degree of message loss and node failures as well. The drawback of the algorithm is the large number of packets transmitted in the network, referred to as broadcast storm: whether it is necessary or not, each node will retransmit each message. In a dense network the unnecessarily high number of messages can cause frequent collisions (and thus performance degradation) and it wastes network energy as well [19].

To reduce the number of routing messages, probabilistic variations of the flood routing were proposed. The main idea behind these algorithms is that a node randomly decides whether to forward a message or not. Several variants of random flood algorithms were proposed. According to the simplest protocol, a node will rebroadcast a message after its first reception with probability $p$ and discards it with probability $1-p$ (clearly, $p=1$ results in basic flooding). Test results verify what intuition suggests: higher $p$ values provide higher network coverage than small $p$ values. Moreover,

978-83-60810-14-9/08/$25.00 © 2008 IEEE

in sufficiently large and sufficiently dense random networks there is an interesting bimodal behavior of the algorithm: if $p > p_c$ , where the critical probability $p_c$ depends on the network topology, the message reaches practically all nodes in the network, otherwise only a small portion of the network receives the message. Thus the optimal choice clearly would be $p = p_c$ . Modified algorithms try to further increase the performance by various modifications: e.g. premature message death can be avoided by varying *p* as a function of hop-distance from the source: nodes close to the source rebroadcast messages with higher probability, while distant nodes use smaller *p*. A comprehensive study on random flooding algorithms can be found in [15]. The common problem with these algorithms is that the design parameters (probabilities) depend on actual network layout, no automatic or adaptive solution is known. Especially in a network with varying node density the probabilistic algorithms tuned to work reliably will be suboptimal.

More sophisticated adaptive schemes can successfully handle networks with varying properties as well. Location based algorithms use node positions to optimize retransmissions [20], graph-based algorithms utilize connectivity information up to 2-hop neighbors to construct a dominating set in a distributed way [21]. Other heuristic rules, e.g. message counters and neighbor coverage schemes were also proposed [19]. These schemes can adapt automatically to changing topology at the price of higher complexity.

Sensor networks are often modeled by the Poisson Boolean model. In this model nodes are scattered on the plane by a Poisson point process with constant density $\lambda$, and each node has a fixed communication radius $r$ (disc model). Two nodes can communicate with each other if their distance is less than $r$ . The Poisson point process applies reasonably well to controlled, but still random deployments. The simplistic disc model, however, is far from reality, where communication range is not circularly symmetric, but rather has an anisotropic shape. Speaking of connectivity, however, the disc model has an important property: it can be considered the worst case model, since other transmission models provide easier percolation under similar circumstances [11].

Percolation theory [22] has important impact on wireless communication networks. Results in continuum percolation theory show that if the density of transmitting stations (or alternatively, their communication power) in a Poisson Boolean network is greater than a critical value $\lambda_c$ then an unbounded connected component is formed with probability one (the network percolates) [23], [24]. On the other hand, if $\lambda < \lambda_c$ then all connected components are finite with probability one. Thus percolation is an important property of a network if long distance multi hop communication is required.

In this paper a new percolation-inspired solution will be proposed based on probabilistic flood routing algorithms. The new algorithm adaptively sets the rebroadcast probability based on the network topology, using locally available neighborhood information only. The proposed algorithm provides high performance with a low number of messages and can adapt its behavior dynamically to the network properties. The proposed algorithm is only marginally more complex than the basic flood routing thus it can be successfully used in applications with limited resources as well.

The rest of the paper is organized as follows. In Section II related works will be summarized. In Section III the proposed algorithm will be introduced. An important property of the algorithm will be proven based on percolation theory: every message broadcast in the network will reach infinite number of nodes almost surely. In practice this property ensures that the message reaches almost all of the nodes in a finite-sized network. The performance evaluation of the algorithm through simulation can be found in Section IV. Conclusions are drawn in Section V.

## II. Related Work

Results of percolation theory have been applied to wireless networks, forming the theory of Continuum Percolation [25]. More realistic network models, i.e. unreliable communication links and anisotropic radiation patterns were studied in [23]. An important result of [23] shows that the debatable disc model actually provides a conservative estimate of the network connectivity: percolation in networks with disc communication shapes is more difficult than in networks with any other convex communication shape with the same surface (“discs are hardest” conjecture). This property justifies the usage of the disc model when connectivity issues are studied.

Results from percolation theory have been infiltrating various recent sensor networking algorithms. As a practical application of this phenomenon, the distributed minimum-link-degree rule was proposed to counterbalance local spatial inhomogeneities in ad-hoc networks [24]. To provide network-wide connectivity, the transmission powers of the nodes are tuned so that each node has at least a minimum number of neighbors.

In [26] a distributed energy saving mechanism is used by switching the nodes in the network on and off, while providing percolation in the network. The simple rule for the algorithm is to keep the active time ratio of the nodes larger than $\lambda / \lambda_c$ . This algorithm does not scale well since each node in the network has to know the global parameter $\lambda$ . The idea was further elaborated in [27], where a practically useable distributed algorithm was proposed for the control of the active time ratio. In this algorithm only the local density, namely the number of neighbors is used. For a node with degree $k$ the active ratio $\eta_k$ is defined as follows:

$$\eta_k = \begin{cases} \dfrac{\varphi}{k} & k > \varphi \\ 1 & k \leq \varphi \end{cases} ,$$

where $\varphi$ is a density-independent design parameter. A similar idea will be used in the proposed flooding algorithm.

## III. Percolation-driven flooding

The idea behind the percolation-driven flood routing is the following: Use probabilistic flooding and set the retransmission probability adaptively at every node so that with high probability the message reaches almost all of the nodes in the network. The algorithm is the following:

1. The source node broadcasts the message with probability one.

2. After the first reception of a message node $n$ rebroadcasts it with probability $p_n$, and discards it with probability $1-p_n$. Copies of the same message received multiple times are discarded with probability one. Probability $p_n$ is defined as

$$p_n=\begin{cases}\frac{K_{\min}}{K_n} & K_n>K_{\min}\\ 1 & otherwise\end{cases} \qquad (1)$$

where $K_n$ is the degree of node $n$ (i.e. the number of neighbors of node $n$ ), and the $K_{\min}$ design parameter is the required minimum number of neighbors. Two nodes are neighbors if they can hear each other (i.e. a symmetric link exists between them).

The algorithm requires neighbor discovery to determine the number of neighbors. Each node can gather this local information by transmitting and receiving HELLO messages. In practical situations parameter $K_{\min}$ is chosen around 7, as will be justified by the simulation examples.

*Theorem* 1: If in the infinite random network, generated by a Poisson Boolean process with density $\lambda$ and communication radius r, there exists design parameter $K_{\min}<\infty$ independent of $\lambda$, such that when node n with degree $K_n$ transmits a message with probability according to (1), then each message reaches infinite number of nodes with probability one.

Proof: Let us denote the graph representing the network by $G(\lambda,r,1)$, where $\lambda$ is the density, r is the communication radius, and the third parameter is a probabilistic value used in the generator process (used and discussed later) and is simply set to one at the moment. Using scaling, we define another process with density $\lambda'=\lambda r^2$ and communication radius 1. This process yields the same graph $G(\lambda',1,1)$. Now let us use the following Theorem 2 [27]:

*Theorem* 2: Given $G(\lambda',1,1)$ with $\lambda'>\lambda'_c$, there exists $2<\phi<\infty$, independent of $\lambda$, such that when each node with degree k is active with probability

$$p(k)=\begin{cases}\frac{\phi}{k} & k>\phi\\ 1 & k\leq\phi\end{cases},$$

then $G(\lambda',1,p(.))$ is percolated. The proof of Theorem 2 can be found in [27].

If in the network we decide *before* a particular message is broadcasted which nodes will retransmit the message (when they receive it) by activating/deactivating nodes with probability (1) we get $G(\lambda',1,p(.))$. In this way we construct a virtual network for each message and we can apply Theorem 2 to it. Thus there exist $K_{\min}=\phi<\infty$ so that $G$ percolates. From this it follows that the message will reach infinite number of nodes [22]. ■

In real-world finite-size networks Theorem 1 means that with an appropriate choice of $K_{\min}$ each message reaches practically all nodes in the network. Further properties of the algorithm will be analyzed through simulation examples.

## IV. Performance Evaluation

In this section the performance of the algorithm will be evaluated through simulation results. First the performance metrics and the network/communication models will be introduced, followed by the simulation environment and the simulation scenarios. Finally, the results of the simulations will be presented to illustrate the behavior of the proposed algorithm.

### A. Performance metrics

The main performance criteria of broadcast algorithms are the message delivery ratio and the number of sent messages. If the number of nodes in the network is N and the number of nodes receiving the message is $N_{rec}$ then the coverage ratio is defined as $C_r=\frac{N_{rec}}{N}$.

Clearly, $C_r$ close to 1 is required for a good quality of service.

Another quality metric is the total number of sent packets M while a message is propagated in the network. Trivially, for basic flood routing $M=N$, if $C_r$ = 1. We expect lower number of messages and still high delivery ratio for a good performance algorithm. For easier comparison, we will use the normalized number of messages defined as

$$M_{norm}=\frac{M}{N}.$$

If flooding is used for route discovery, an important performance metric is the length $L$ of the discovered route.

Other possible metrics are message delay/latency, length of flood period, and number of collisions. The actual low level details, e.g. implementation, hardware and MAC layer properties are not investigated in detail in this paper, but their effect is studied through high level parameters they effect, i.e. $C_r$, $M_{norm}$, and $L$ will be studied.

### B. Network model

To model random deployment of nodes or possibly mobile networks, in the simulations nodes are placed at random, according to a uniform distribution on a two-dimensional area. Finite communication distances are represented by the disc model, where a communication link is assumed between two nodes when they are less than the communication radius apart. According to the results published in [23], this – otherwise rather simple and idealistic – model can be considered a worst case model.

Our communication channel model does not deal with specific details of the physical layer or the MAC-layer; rather we use a probabilistic model: a message can reach a neighbor within the communication radius with probability $p_{rec}<1$. This high-level model represents message losses due to collision, fading, or other disturbances as well. The model does not distinguish between individual phenomena but rather incorporates the different sources in one parameter, thus some aspects of reality (e.g. inter-message dependencies) are neglected, but the model is faithful enough to provide useful and easy means for testing.

### *C. Simulation environment*

To perform high speed simulation, a simulator was written in C to validate the efficiency of the proposed algorithm. The program places nodes randomly according to the different test scenarios considered. The number of nodes *N*, the communication radius *r*, and $K_{\min}$ are input parameters.

In each simulation, a new placement is generated, and a source node starts transmission. The simulator returns the size of connected component containing the source node, the number of nodes receiving the packet, and the total number of messages. Optionally, the software can visualize the topology of the network, the active data paths, and the nodes' reception status.

### *D. Test scenarios*

In the test we used 2 different scenarios to model sensor network setups. The first scenario is a random uniform distribution (with constant density), while in the second scenario we used three regions; in each region nodes were placed with uniform random distribution, but with different densities. Typical examples for the test networks can be seen in Fig. 1 and Fig. 2.

### *E. Test results*

To test coverage ratio $C_r$ and the number of messages $M$ we placed 3000 nodes in a square-shaped area, as shown in Fig. 1. We varied the communication radius to provide different network densities: the average number of neighbors *K* was set to 10, 30, and 100. The $K_{\min}$ value varied from 1 to 20 in 0.25 steps. We run the simulation 100 times for each communication radius, $K_{\min}$, and $p_{rec}$ values, thus each point in the graph is the average of 100 experiments.

Fig. 3 shows coverage vs. $K_{\min}$, while $p_{rec}$ was set to constant 1. The graphs for different network densities are similar: When $K_{\min}$ is low ( $K_{\min}<3$ ), the message delivery ratio is very low. If $K_{\min}$ is increased, the coverage is increasing very quickly, the critical value being around $K_{\min}\approx 4.5$ . Coverage reaches 90% at $K_{\min}\approx 5$ . If $K_{\min}>7$ , practically all nodes in the network receive the message. Fig. 3 shows that percolation driven flood can indeed be used in networks with different densities: $K_{\min}$ is independent of the actual network density.

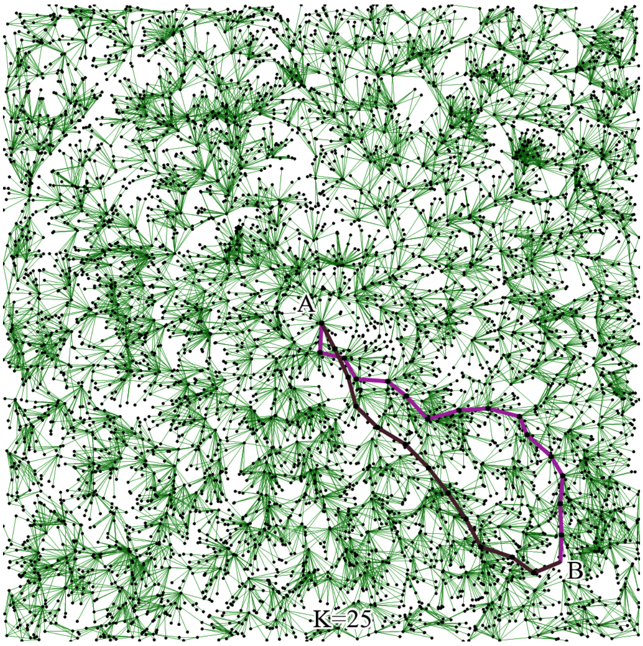


Figure 1: Uniform random distribution scenario with 3000 nodes. The average number of neighbors is 25. Black and purple lines show examples for the message paths between nodes A and B, discovered by flood and percolation driven flood, respectively.

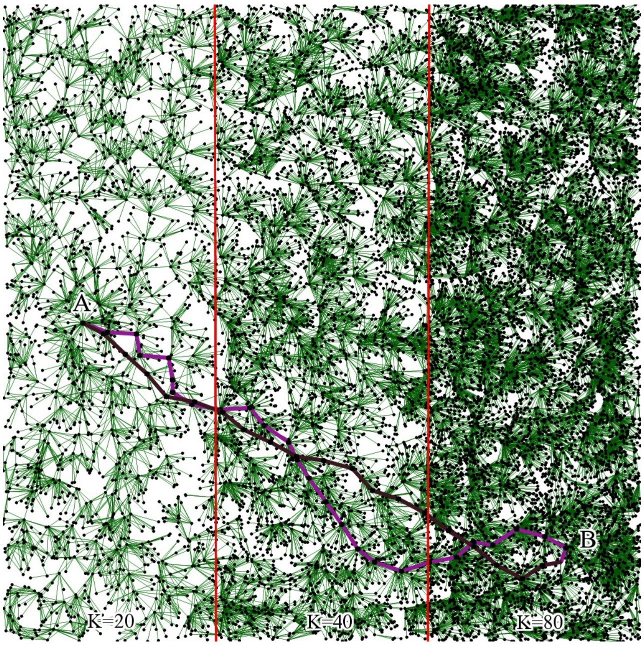


Figure 2: The 3-density scenario. The average numbers of neighbors in the regions from left to right are 20, 40, and 80. Black and purple lines show examples for the message paths between nodes A and B, discovered by flood and percolation driven flood, respectively.

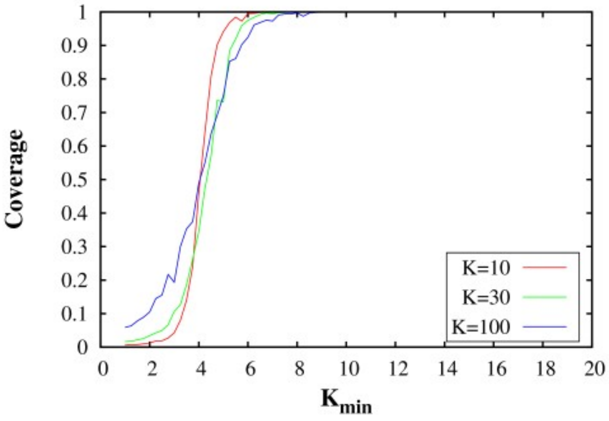


Figure 3: Coverage vs. $K_{min}$ for different network densities in the uniform distribution scenario (constant network density). $P_{rec}$ = 1.

In Fig. 4 the normalized number of sent messages $M_{norm}$ is shown vs. $K_{\min}$, for the previous experiments. In case of the conventional flood routing $M_{norm}=1$, because all nodes relay the received message. As Fig. 3 shows, $K_{\min}>7$ gives almost perfect delivery ratio. Around this value $K_{\min}\approx 7$ the total number of messages is greatly decreased, as shown in Fig. 4. Depending on the node density, in the tests the total number of messages were reduced by 30..95%, higher reduction rates belonging to higher densities.

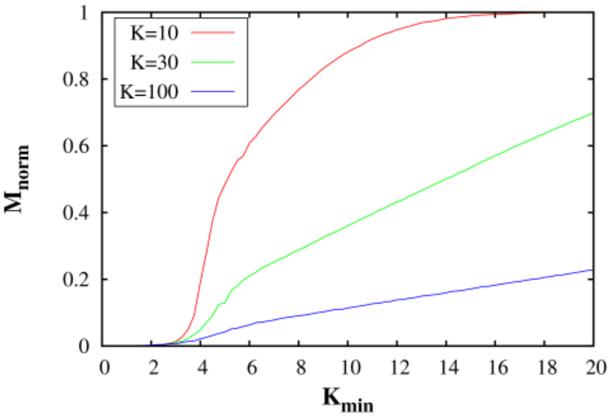


Figure 4: Total number of messages vs. Kmin for different network densities in the uniform distribution scenario (constant network density). Prec = 1.

Clearly, in networks, where the average node degree $K$ is only slightly higher than $K_{\min}$ the number of messages can be decreased only moderately, while in dense networks a much lower number of messages is enough to provide good delivery ratio, as shown in Fig. 4. According to Fig. 3 and Fig. 4, in dense networks the percolation driven flood algorithm can effectively reduce the number of message while maintaining good coverage.

Fig. 5 shows the effect of unreliable communication links. In the experiment a constant density (K=30) was used, while the $p_{rec}$ reception probability was varied between 0.3 and 1. The figure shows that the algorithm can provide high coverage even in the presence of bad quality communication links, but as intuitively expected, higher $K_{\min}$ is necessary as the communication channel degrades. Fig. 6 shows the associated number of messages for the unreliable communication experiment.

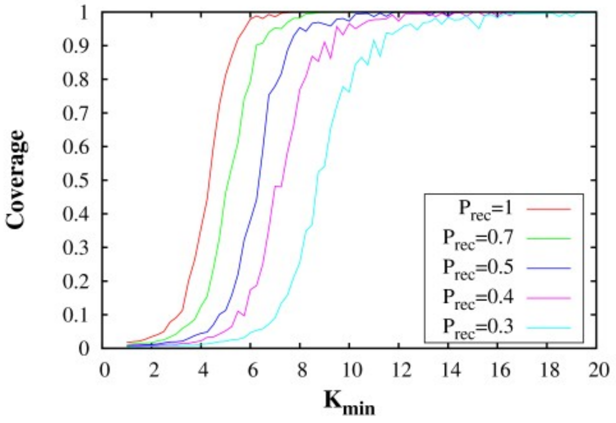


Figure 5: Coverage vs. Kmin for different prec in the uniform distribution scenario (constant network density). K = 30

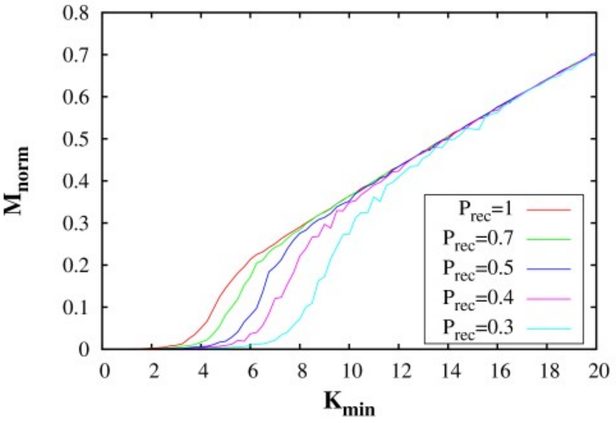


Figure 6: Number of total messages vs. for different prec in the uniform distribution scenario (constant network density). K = 30.

Percolation driven flood routing algorithm is capable of handling varying network densities. To test this property we used the scenario illustrated in Fig. 2, where three areas with different densities are present. Clearly, a constant retransmission probability would either be suboptimal (set to provide sufficient coverage in the less dense area as well) or would not provide good coverage in the sparse regions of the network. Fig. 7. illustrates the effectiveness of the percolation driven flood algorithm, showing the coverage vs. $K_{\min}$ when the densities of regions from left to right were set to [10, 20, 40], [30, 60, 120], and [100, 200, 400]; and $p_{rec}=1$. The behavior is quite similar to that of the first scenario (constant density): percolation happens around the same $K_{\min}\approx 5$, and practically full coverage can be provided if $K_{\min}>7$. The associated total numbers of messages are shown in Fig. 8, where the gain with respect to the basic flooding algorithm is apparent.

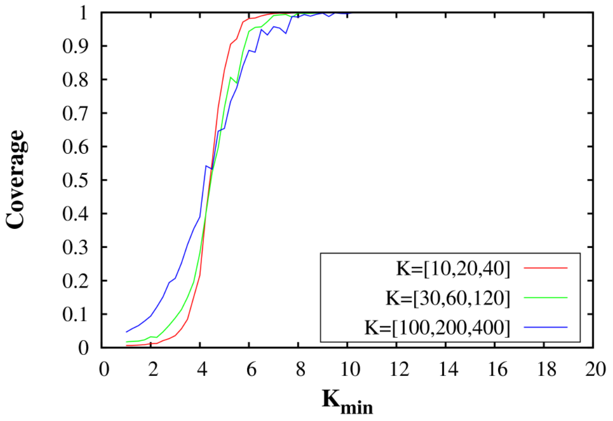


Figure 7: Coverage vs. Kmin for different network densities in the varying network density scenario. Prec = 1.

The length $L$ of the discovered route is important when flooding is used for route discovery. To test the properties of percolation driven flooding in this respect we measured hop-distances discovered by flooding and percolation driven flooding, for different $K_{\min}$ values. In the tests the uniform distribution (constant density) scenario was used with $K=35$ and $p_{rec}=1$ . The source node was placed to the center of the field and hop distances to all other nodes were measured. The histogram created from averaging 100 independent experiments is shown in Fig. 9.

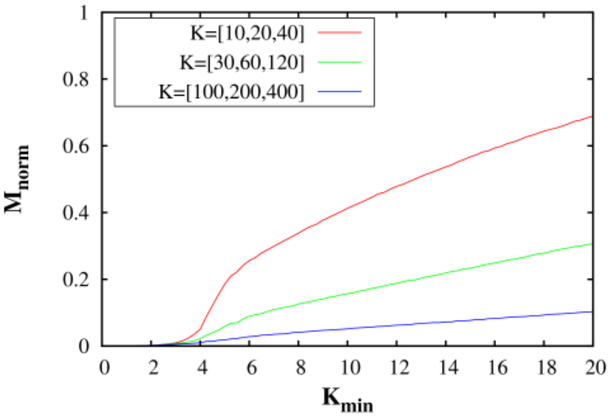


Figure 8: Number of total messages vs. Kmin for different network densities in the varying network density scenario. Prec = 1

The distribution is linear for small hop distances. The explanation is shown in Fig. 10 for the ideal case: messages are propagated in belt-shaped increments. The areas of the subsequent belts increase linearly, thus the number of nodes in each belt is linearly increasing. If the node density is smaller, the number of nodes in each belt is smaller, thus the slope of the histogram will be smaller. According to Fig. 9, smaller $K_{\min}$ also causes decrease in the slope: e.g. for $K_{\min}=9$ , the slope is 40% smaller than in the basic flooding case. This means that the detected route is 40% longer than that of the basic flooding.

The declining part at higher hop distances is due to the finite size of the network: the messages reach the network perimeter and eventually die. The tail of the distribution shows the maximum hop distance from the source, which is in the square setup ideally $\sqrt{2}$ times the hop distance connected to the maximum of the histogram.

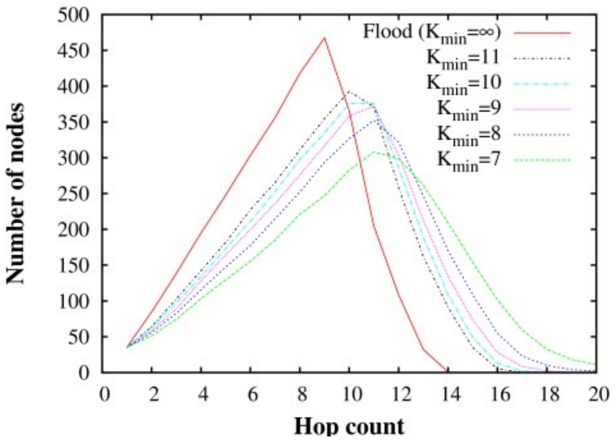


Figure 9: Distribution of the hop counts in the uniform distribution network for different Kmin values. Prec = 1.

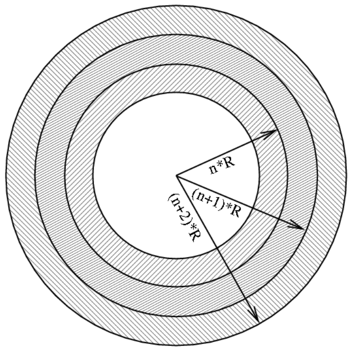


Figure 10: Ideal propagation of a message in a dense network.

The effect of unreliable links is shown in Fig. 11. The histogram shows the average of 100 experiments conducted in the uniform test scenario with K=35, and $p_{rec}=0.4..1$ . As intuitively expected, unreliable links cause smaller slope, i.e. higher hop distances. E.g., link quality $P_{rec}=0.7$ results in the increase of hop distance by 18%, while for the rather unreliable link quality $P_{rec}=0.4$ the increase was as high as 63%.

As simulation examples show, percolation-driven flooding results in considerably higher hop distances than basic flood routing when small $K_{\min}$ parameters are used. Naturally, low link quality also causes larger hop distances. This effect may be an undesired side effect in route discovery protocols, thus in these applications basic flooding may be more advantageous.

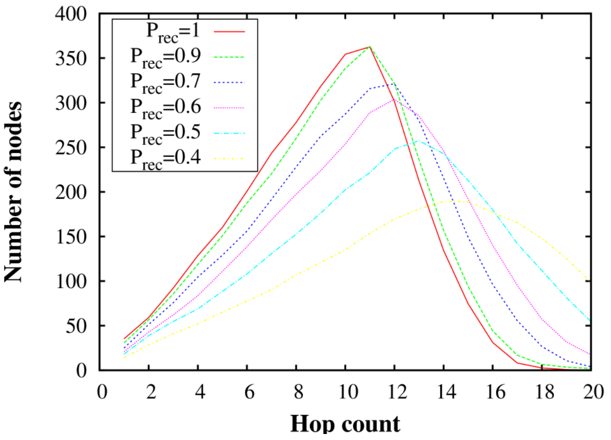


Figure 11: Distribution of the hop counts in the uniform distribution network for different $P_{rec}$ values. $K_{min}$ = 9

## V. Conclusion

A percolation driven flood routing algorithm was proposed, which uses only locally available neighborhood information to reduce broadcast storm. The algorithm is able to massively reduce the number of messages in the network and maintaining high delivery ratio at the same time. Theoretical results prove the usefulness of the algorithm: it is able to provide high coverage, if the network density is high enough.

Simulation tests were performed to validate the performance of the algorithm. The proposed algorithm reduced the total number of messages in the network by 30-95%, depending on the network density, while the coverage was almost 100%, with appropriate choice of $K_{\text{min}}$ $(K_{\text{min}}>7)$. The percolation driven flood algorithm is adaptive to changing node density thus provides high coverage with low number of messages in all scenarios. The algorithm is robust in the presence of unreliable links as well.

According to test result, the percolation-driven flood routing algorithm results considerably (even 50%) longer hop-distances when used for route discovery purposes.

The overhead of the algorithm is very low since only the number of neighbors must be known by each node. This information can be gained by simple neighborhood discovery protocols, e.g. sending and receiving HELLO messages.

For route discovery purposes the application of the traditional flooding is more advantageous, if the side effect of the longer routes is undesirable. On the other hand, the proposed algorithm is a superior and robust alternative to traditional flooding algorithm in dense networks for message dissemination purposes.

## References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in Proc. of WSNA, pages 88-97, 2002.

[2] A. Arora, et al., “A line in the sand: a wireless sensor network for target detection, classification, and tracking,” *Comput. Networks* vol. 46, Dec. 2004, pp. 605-634.

[3] A. Ledeczi, et al., “Countersniper system for urban warfare,” *ACM Trans. Sen. Netw.* Vol. 1, Nov. 2005) pp. 153-177.

[4] T. He, et al., “VigilNet: An integrated sensor network system for energy-efficient surveillance,” *ACM Trans. Sen. Netw.* Vol. 2, Feb. 2006, pp. 1-38.

[5] B. Karp and H. T. Kung, “Greedy perimeter stateless routing (GPSR) for wireless networks,” in Proc. ACM MobiCom, 2000, pp. 243–254.

[6] Y. B. Ko and N. H. Vaidya,“Location-aided routing (LAR) in mobile ad hoc networks” in Proc. ACM MobiCom, 1998, pp. 66–75.

[7] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward,“A distance routing effect algorithm for mobility (DREAM)” in Proc. ACM MobiCom, 1998, pp. 76–84.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Trans. Netw*. Vol 11, pp. 2–16, Feb. 2003.

[9] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” in Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[10] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”. Boston, MA: Kluwer Academic, 1996.

[11] V. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in Proc. IEEE INFOCOM, Apr. 1997, pp. 1405–1413.

[12] Z. Haas and M. Pearlman, “The performance of query control schemes for the zone routing protocol,” in Proc. ACM SIGCOMM, 1998, pp. 167–177.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," in the Proceeding of the Hawaii International Conference System Sciences, Hawaii, January 2000.

[14] A. Manjeshwar and D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks," in the Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[15] Z. J. Haas, J.Y., Halpern, L. Li, “Gossip-based ad hoc routing.” *IEEE/ACM Trans. Netw*. Vol. 14, pp. 479–491, Jun. 2006.

[16] M. Maróti, “Directed flood-routing framework for wireless sensor networks,” in Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware, pp. 99–114.

[17] L. H. Costa, M. D. De Amorim, S. Fdida, “Reducing latency and overhead of route repair with controlled flooding,” *Wirel. Netw*. Vol. 10, Jul. 2004.

[18] P. Gburzynski and W. Olesinski. “On a Practical Approach to Low-cost Ad hoc Wireless Networking” *Journal of Telecommunications and Information Technology*, no. 1, 2008, pp. 29-42

[19] Zigbee Alliance, online: http://www.zigbee.org

[20] Y. C. Tseng, S. Y. Ni, and E. Y. Shih, “Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network,” in Proceedings of the 21st International Conference on Distributed Computing Systems, 2001, pp. 481–488.

[21] J.Yang, B. Kim, M-T. Sun, T-H. Lai, “Location-Aided Broadcast in Wireless Ad Hoc Networks,” *Journal of Information Science and Engineering*, vol. 23, pp. 869–884, 2007.

[22] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, “On calculating power aware connected dominating set for efficient routing in ad hoc wireless networks,” *Journal of Communications and Networks*, Vol. 5, 2002, pp. 169–178.

[23] G. Grimmet, Percolation. Springer-Verlag, New York, 1989.

[24] M. Franceschetti, L. Booth, M. Cook, R. Meester, J. Bruck, “Percolation in Multi-hop Wireless Networks” In Caltech Paradise, ETR055, September 2, 2003.

[25] I. Glauche, W. Krause, R. Sollacher M. Greiner, “Continuum percolation of wireless ad hoc communication networks,” *Physica A*, vol. 325, pp. 577–600, 2003

[26] J. Jonasson, ”Optimization of shape in continuum percolation,” *Annals of Probability* 29(2), 624-635, 2001.

[27] O. Dousse, P. Mannersalo, P. Thiran, “Latency of wireless sensor networks with uncoordinated power saving mechanisms,”.in *Proceedings of the 5th ACM international Symposium on Mobile Ad Hoc Networking and Computing*, 2004, MobiHoc '04, pp. 109-120, 2004.

[28] Z. Kong, E. M. Yeh, “Distributed energy management algorithm for large-scale wireless sensor networks,” in Proceedings of the 8th ACM international Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '07, pp. 209-218, 2007.

# Workshop on Computational Optimization

MANY real world problems arising in engineering, economics, medicine and other domains can be formulated as optimization tasks. These problems are frequently characterized by non-convex, non-differentiable, discontinuous, noisy or dynamic objective functions and constraints which ask for adequate computational methods.

The aim of this workshop is to stimulate the communication between researchers working on different fields of optimization and practitioners which need reliable and efficient computational optimization methods.

We invite original contributions related with both theoretical and practical aspects of optimization methods.

The list of topics includes, but is not limited to:

- unconstrained and constrained optimization
- combinatorial optimization
- global optimization
- multiobjective and multimodal optimization
- dynamic and noisy optimization
- large scale optimization
- parallel and distributed approaches in optimization
- random search algorithms, simulated annealing, tabu search and other derivative free optimization methods
- interval methods
- nature inspired optimization methods (evolutionary algorithms, ant colony optimization, particle swarm optimization, immune artificial systems etc)
- hybrid optimization algorithms involving natural computing techniques and other global and local optimization methods
- memetic algorithms
- optimization methods for learning processes and data mining
- computational optimization methods in statistics, econometrics, finance, physics, medicine, biology etc.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 887–891

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Ant Colony System Approach for Protein Folding

Stefka Fidanova, Ivan Lirkov
Institute for Parallel Processing, Bulgarian Academy of Sciences, Acad G. Bonchev, bl. 25A, 1113 Sofia, Bulgaria
E-mail: stefka@parallel.bas.bg, ivan@parallel.bas.bg
http://parallel.bas.bg/~stefka/, http://parallel.bas.bg/~ivan/

***Abstract*—The protein folding problem is a fundamental problem in computational molecular biology and biochemical physics. The high resolution 3D structure of a protein is the key to the understanding and manipulating of its biochemical and cellular functions. All information necessary to fold a protein to its native structure is contained in its amino-acid sequence. Even under simplified models, the problem is NP-hard and the standard computational approach are not powerful enough to search for the correct structure in the huge conformation space. Due to the complexity of the protein folding problem simplified models such as hydrophobic-polar (HP) model have become one of the major tools for studying protein structure. Various optimization methods have been applied on folding problem including Monte Carlo methods, evolutionary algorithm, ant colony optimization algorithm. In this work we develop an ant algorithm for 3D HP protein folding problem. It is based on very simple design choices in particular with respect to the solution components reinforced in the pheromone matrix. The achieved results are compared favorably with specialized state-of-the-art methods for this problem. Our empirical results indicate that our rather simple ant algorithm outperforms the existing results for standard benchmark instances from the literature. Furthermore, we compare our folding results with proteins with known folding.**

***Index Terms*—Ant Colony Optimization, metaheuristics, hydrophobic-polar model, protein folding**

## I. Introduction

THE number of amino acids and their sequence give a protein its individual characteristics. The number of amino acids in each protein ranges approximately between 20 and 40000, although most proteins are around hundred amino acids in length. Each protein's sequence of amino acids determines how it folds into a unique three dimensional structure that is its minimum energy state. Knowledge of 3D structure of proteins is crucial to pharmacology and medical sciences for the following important reasons. Most drugs work by attaching themselves to a protein so that they can either stabilize the normally folded structure or disrupt the folding pathway, which leads to a harmful protein. Thus, knowing exact 3D shapes will help to design drugs.

Determining the functionality of a protein molecule from amino acid sequence remains a central problem in computational biology, molecular biology, biochemistry, and physics. A system of differential equations is used to describe the forces, which affect the folding. It is very complicate and difficult to be solved. Even the experimental determination of these conformations is often difficult and time consuming. It is common practice to use models that simplify the search space of possible conformation. The aim is to find a conformation, which is close to the real one and than it to be specify using system of differential equations. So, as closer is the conformation, as less complex is the system of differential equations. Thus the computational time decreases. These models try to generally reflect different global characteristics of protein structures. In the hydrophobic-polar (HP) model [4] the primary amino acid sequence of a protein (which can be represented as a string over twenty-letter alphabet) is abstracted to a sequence of hydrophobic (H) and polar (P) residues that is represented as a string over the letter H and P. It describes the proteins based on the fact that hydrophobic amino acids tend to be less exposed to the aqueous solvent than the polar ones, thus resulting in the formation of a hydrophobic core in the spatial structure. In the model, the amino acid sequence is abstracted to a binary sequence of monomers that are either hydrophobic or polar. The structure is a chain whose monomers are on the vertices's of a three dimensional cubic lattice. The free energy of a conformation is defined as the negative number of non-consecutive hydrophobic-hydrophobic contacts. A contact is defined as two non-consecutive monomers in the chain occupying adjacent sites in the lattice. In spite of its apparent simplicity, finding optimal structures of the HP model on a cubic lattice is NP-complete problem [2].

Ant Colony Optimization (ACO) is a population-based stochastic search method for solving a wide range of combinatorial optimization problems. ACO is based on the concept of indirect communication between members of a population through interaction with the environment. Ants indirectly communicate with each other by depositing pheromone trails on the ground and thereby influencing the decision processes of other ants. From the computational point of view, ACO is an iterative construction search method in which a population of simple agents (ants) repeatedly constructs candidate solutions to a given problem. This construction process is probabilistically guided by heuristic information on the given problem instances as well as by a shared memory containing experience gathered by the ants in previous iterations.

This work is an investigation of the HP model in a three dimensional cubic lattice using an ACO as a tool to find the optimal conformation for a given sequence. The achieved results are evaluated and compared with other metaheuristic methods using 10 sequences of 48 monomers from the literature and with real proteins with known folding.

The paper is organized as follows: the problem is described in section 2. The ACO algorithm is explained

978-83-60810-14-9/08/$25.00 © 2008 IEEE

in section 3. The achieved results are discussed in section 4. The paper ends with a summary of the conclusions.

## II. The Protein Folding Problem

Efforts to solve the protein folding problem have traditionally been rooted in two schools of thought. One is based on the principles of physics: that is, the thermodynamic hypothesis, according to which the native structure of the protein corresponds to the global minimum of its free energy. The other school of thought is based on the principles of evolution. Thus methods have been developed to map the sequence of one protein (target) to the structure of another protein (template), to model the overall fold of the target based on that of the template and to infer how the target structure will be changed, related to the template, as a result of substitutions [1].

Accordingly methods for protein-structure prediction has been divided into two classes: de novo modeling and comparative modeling. The de novo approaches can be further subdivided, those based exclusively on the physics of the interactions within the polypeptide chain and between the polypeptide and solvent, using heuristic methods [9], [10], and knowledge-based methods that utilize statistical potential based on the analysis of recurrent patterns in known protein structures and sequences. The comparative modeling models structure by copying the coordinates of the templates in the aligned core regions. The variable regions are modeled by taking fragments with similar sequences from a database [1].

The processes involving in folding of proteins are very complex and only partially understood, thus the simplified models like Dill's HP model have become one of the major tools for studying proteins [4]. The HP model is based on the observation that hydrophobic interconnection is the driving force for protein folding and the hydrophobicity of amino acids is the main force for development of native conformation of small globular proteins. In the HP model, the primary amino acid sequence of a protein is abstracted to a sequence of hydrophobic (H) and polar (P) residues, amino acid components. The protein conformations of this sequence are restricted to self-avoiding paths on 3 dimensional sequence lattice. One of the most common approaches to protein structure prediction is based on the thermodynamic hypothesis which states that the native state of the protein is the one with lowest Gibbs free energy. In the HP model, the energy of a conformation is defined as a number of topological contacts between hydrophobic amino acid that are not neighbors in the given sequence. More specifically a conformation $c$ with exactly $n$ such H-H contacts has free energy $E(c) = n.(-1)$. The 3D HP protein folding problem can be formally defined as follows. Given an amino acid sequence $s = s_1 s_2 \dots s_n$, find an energy minimizing conformation of $s$, i.e. find $c^s \in C(s)$ such that $E^s = E(c^s) = \min_{c \in C(s)} E(c)$, where $C(s)$ is the set of all valid conformations for s. It was proved that this problem is NP-hard [2].

A number of well-known heuristic optimization methods have been applied to the 3D protein folding problem including Evolutionary Algorithm (EA) [9], Monte Carlo (MC) algorithm [10] and Ant Colony Optimization (ACO) algorithm [7]. An early application of EA to protein structure prediction was presented by Unger and Moult [11]. Their EA incorporates characteristics of Monte Carlo methods. Currently among the best known algorithms for the HP protein folding problem is Pruned-Enriched Rosenblum Method (PERM) [8]. Among these methods are the Hydrophobic Zipper (HZ) method [5] and the Constraint-based Hydrophobic Core Construction Method (CHCCM) [12]. The Core-direct chain Growth method (CG) [3] biases construction towards finding a good hydrophobic core by using a specifically designed heuristic function.

## III. ACO Algorithm for Protein Folding Problem

Real ants foraging for food lay down quantities of pheromone (chemical cues) marking the path that they follow. An isolated ant moves essentially at random but an ant encountering a previously laid pheromone will detect it and decide to follow it with high probability and therefore reinforce it with a future quantity of pheromone. The repetition of the above mechanism represents the auto-catalytic behavior of real ant colony where the more the ants follow a trail, the more attractive that trail becomes.

The ACO algorithm uses a colony of artificial ants that behave as co-operative agents in a mathematical space where they are allowed to search and reinforce path ways (solutions) in order to find the optimal ones. The problem is represented by graph and the ants walk on the graph to construct solutions. After initialization of the pheromone trails, ants construct feasible solutions and the pheromone trails are updated. At each step ants compute a set of feasible moves and select the best one (according to some probabilistic rules) to carry out the rest of the tour. The transition probability is based on the heuristic information and pheromone trail level of the move. The higher the value of the pheromone and the heuristic information, the more profitable is to select this move and resume the search. In the beginning, the initial pheromone level is set to a small positive constant value $\tau_0$ and then ants update this value after completing the construction stage. ACO algorithms adopt different criteria to update the pheromone level. In our implementation Ant Colony System (ACS) approach is used [6]. In ACS the pheromone updating consists of two stages: local update and global update. While ants build their solutions, at the same time they locally update the pheromone level of the visited paths by applying the local update rule as follows:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\tau_0 \tag{1}$$

Where $\tau_{ij}$ is an amount of the pheromone on the arc $(i, j)$ of the 3D cube lattice, $\rho$ is a persistence of the trail and the term $(1-\rho)$ can be interpreted as trail evaporation. The aim of the local update rule is to make better use of the pheromone information by dynamically changing the desirability of edges. Using this rule, ants will search in a wide neighborhood of the best previous solution. As is shown in the formula, the

TABLE I
STANDARD BENCHMARK INSTANCES

| | |
|---|---|
| 1 | HPHHPPHHHHPHHHPPHHPPHPHHPHPHHPPHHPPPHPPPPPPPPHHP |
| 2 | HHHHPHHPHHHHHPPHPPHHPPHPPPPPPHPPHPPPHPPHHPPHHHPH |
| 3 | PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP |
| 4 | PHPHHPPHPHHHPPHHPHHPPPHHHHHHPPHPHHPHPHPPPPHPPHPHP |
| 5 | PPHPPPHPHHHHPPHHHHPHHPHHHPPHPHPHPPHPPPPPPHHPHHPH |
| 6 | HHHPPPHHPHPHHPHHPHHPHPPPPPPPHPHPPHPPPHPPHHHHHHPH |
| 7 | PHPPPPHPHHHPHPHHHHPHHPHHPPPHPHPPPHHHPPHHPPHHPPPH |
| 8 | PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPHPHHHPPPPH |
| 9 | PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPHPHHHPPPPH |
| 10 | PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHHPPHHHHHHHPPHH |

pheromone level on the paths is highly related to the value of evaporation parameter $\rho$. The pheromone level will be reduced and this will reduce the chance that the other ants will select the same solution and consequently the search will be more diversified. When all ants have completed their solutions, the pheromone level is updated by applying the global updating rule only on the paths that belong to the best solution since the beginning of the trials as follows:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij}, \tag{2}$$

$$\text{where } \Delta\tau_{ij} = \begin{cases} -E_{gb} & \text{if } (i,j) \in \text{best solution} \\ 0 & otherwise \end{cases}$$

The $E_{gb}$ is the free energy of the best folding. This global updating rule is intended to provide a greater amount of pheromone on the paths of the best solution, thus intensify the search around this solution.

There are six possible positions on the 3D lattice for every amino acid. They are the neighbor positions of the precedence amino acid. Since conformations are rotationally invariant, the position of the first two amino acids can be fixed without loss of generality. During the construction phase, ants fold a protein from the left end of the sequence adding one amino acid at a time based on the two sources of information: pheromone matrix value, which represents previous search experience, and heuristic information. The transition probability to select the position of the next amino acid is given as:

$$P_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{k\in Unused} \tau_{ik}^{\alpha}\eta_{ik}^{\beta}} \tag{3}$$

Where $\tau_{ij}$ is the intensity of the pheromone deposited by each ant on the path $(i, j)$, $\alpha$ is the intensity control parameter, $\eta_{ij}$ is the heuristic information equal to the number of new H-H contacts if the position $j$ is chosen, $\beta$ is the heuristic parameter. Thus the higher the value of $\tau_{ij}$ and $\eta_{ij}$, the more profitable is to put the next amino acid on the position $j$. When the next amino acid is polar, the probability is $P_{ij} = 0$. In this case the position is chosen randomly between allowed positions. When the set of allowed positions is empty, the ant does some steps back and after that it continues construction of the solution. On a Fig. 1 is the ACO algorithm.

```
Ant Colony Optimization

Initialize number of ants;
Initialize the ACO parameters;
while not end-condition do
    for k=0 to number of ants
        ant k starts from random node;
        while solution is not constructed do
            ant k selects a node with probability;
        end while
    end for
    Local search procedure;
    Update-pheromone-trails;
end while
```

Fig. 1. Pseudocode for ACO

## IV. EXPERIMENTAL RESULTS

Ten standard benchmark instances of length 48 for 3D HP protein folding shown in Table I have been widely used in the literature [3], [7], [9]–[11]. Experiments on these standard benchmark instances were conducted by performing a number of independent runs for each problem instance, 20 runs. The following parameter settings are used for all experiment as: $\alpha = \beta = 1$, $\rho = 0.5$. Furthermore, all pheromone values were initialized to $\tau_0 = 0.5$ and a population of 5 ants were used. The algorithm was terminated after 200 iterations. All experiments were performed on IBM ThinkPad Centrino 1.8 GHz CPU, 512 MB RAM running SuSe Linux.

In Table II the achieved results by various heuristic algorithms are compared. For every of the benchmark instances the best found result by various methods is reported.

We compared the solution quality obtained by: hydrophobic zipper (HZ) algorithm [5], the constrain-based hydrophobic core construction (CHCC) method [13], the core-directed chain growth (CG) algorithm [3], the contact interactions (CI) algorithm [11], the pruned-enriched Rosenbluth method (PERM) [7], the ACO algorithm of Hoos (ACO) [10] and the ACS approach presented in this paper. For ACS the best found result and the average result over 20 runs are reported. In the majority of the cases our average results are better

TABLE II
COMPARISON OF 3D PROTEIN FOLDING

| Benchmark | HZ | CHCC | CG | CI | PERM | ACO | ACS | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | best result | average result |
| 1 | 31 | 32 | 32 | 32 | 32 | 32 | 48 | 35.15 |
| 2 | 32 | 34 | 34 | 33 | 34 | 34 | 49 | 36 |
| 3 | 31 | 34 | 34 | 32 | 34 | 34 | 43 | 32.6 |
| 4 | 30 | 33 | 33 | 32 | 33 | 33 | 43 | 30.6 |
| 5 | 30 | 32 | 32 | 32 | 32 | 32 | 43 | 35.15 |
| 6 | 29 | 32 | 32 | 30 | 32 | 32 | 43 | 32.75 |
| 7 | 29 | 32 | 32 | 30 | 32 | 32 | 42 | 33.8 |
| 8 | 29 | 31 | 31 | 30 | 31 | 31 | 42 | 32.95 |
| 9 | 31 | 34 | 33 | 32 | 34 | 34 | 46 | 34.44 |
| 10 | 33 | 33 | 33 | 32 | 33 | 33 | 46 | 36.45 |

than the best found results by other methods. And for all of the cases our best result is better than the best result of other methods. In ACO a local search procedure is used to improve the results. ACS approach is used without local search procedure. The main differences between ACO and ACS implementations are the location of the polar amino acids, the construction of the heuristic information and the pheromone updating. In ACO the authors put the polar amino acids on same direction as precedence amino acid. In our ACS we put the polar amino acids in random way, thus we give to the ants more possibilities in a search process. The main disadvantage of heuristic methods, as it is mentioned by other authors, is that they achieve good folding for short proteins only. For illustration, we compare two real proteins with known folding and the folding achieved by our ACS algorithm, which outperforms others on benchmark tests. Like test problems we choose Hepsidin and c-src Tyrosine Kinase Sh3 Domain (SrcSH3).

The Hepsidin consists of 21 amino acids: GCRFCCNCCP-NMSGCGVCCRP. His folding comprises two crossed sheets and unstructured part between them (see Fig. 2).

The HP representation of the Hepsidin is: HPPHPPPPPH-PHPHPHHPPPH. By our ACS algorithm we achieve the 3D folding represented on Fig. 3. The nodes represent amino acids and the lines represent their succession. We observe two tense, orthogonally situated parts. One of them consists of 3 amino acids and other consists of 4 amino acids. Between them we observe unstructured part. Thus we can conclude that there is high similarity between the real Hepsidin folding and this obtained by our algorithm.

The SrcSH3 protein consists of 62 amino acids. It folding comprise two long parallel situated sheets like a hairpin inside the protein and short sheets at the beginning and at the end of the protein, which are parallel each other and orthogonal to the hairpin, see Fig. 4.

By our ACS algorithm we achieve a folding represented on Fig. 5. The achieved H-H contacts are 19. We observe that there is not similarity between real folding and this achieved by our algorithm. Thus we prove the conclusion of other works [10], that heuristic methods are good for folding short proteins only. Therefore we decide to cut the HP chain of the SrcSH3 protein to short parts consisting of about 10-11 amino acids. We apply our ACS algorithm on every short part and at the end we assemble the folded parts to fold entire protein, see Fig. 6. The achieved H-H contacts are 20. We observe two tense long parallel parts like hairpin. One of them consists of 8 and other 7 amino acids. At one of the ends we observe short tense part orthogonal to the hairpin. Other protein parts are unstructured. Thus we can conclude that there is high similarity with real folding of this protein.

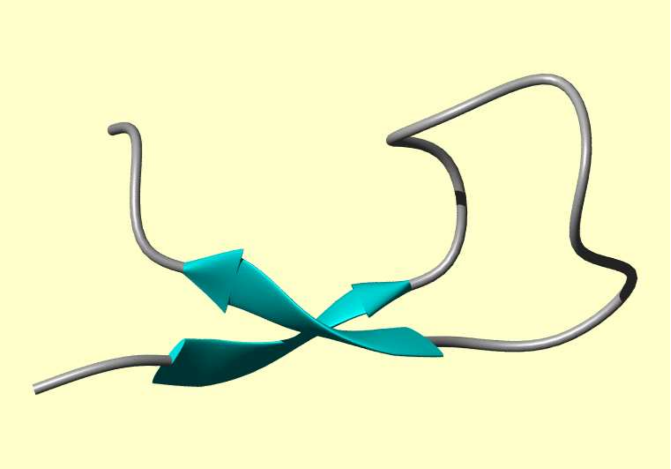


Fig. 2. Hepsidin

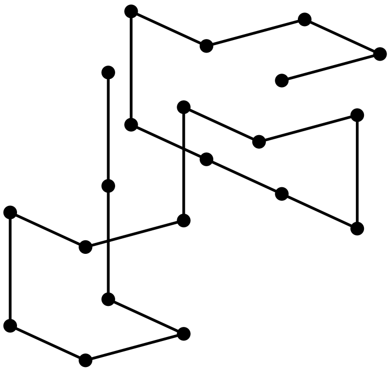


Fig. 3. ACS Hepsidin

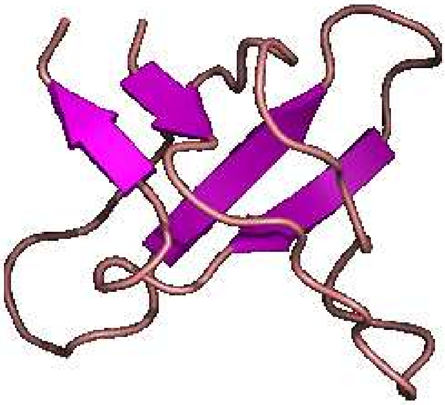


Fig. 4. Tyrosine SrcSH3 folding

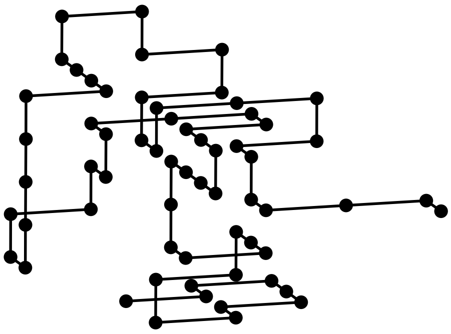


Fig. 5. ACS Tyrosine SrcSH3 folding

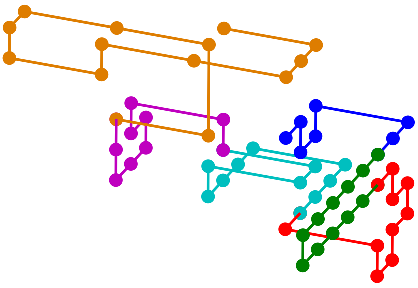


Fig. 6. Partially Tyrosine SrcSH3 folding

## V. Conclusion

Ant Colony System approach can be successfully applied to the 3D protein folding problem. Our algorithm outperforms the weel known from the literature methods. We have shown that the components of the algorithm contribute to its performance. In particular, the performance is affected by the heuristic function and selectivity of pheromone updating. The folding achieved by our algorithm is very similar to the real protein folding when it is applied on short proteins. When the protein is long, first we cut it on short parts, then we apply the algorithm on every one of the parts separately, finally we assemble the protein parts. Thus the achieved folding has high similarity to the real one. The obtained results are encouraging and the ability of the developed algorithm to generate rapidly high-quality solutions can be seen. In the future we will develope and improve the folding algorithm. The aim is to achieve more realistic folding.

## Acknowledgment

S. Fidanova was supported by the Bulgarian Ministry of Education by the grant “Virtual screening and computer modeling for drug design”. This research was partially supported by grant I-1402/2004 from the Bulgarian NSF.

## References

[1] Balev S., Solving the Protein Threading Problem by Lagrangian Relaxation, *Algorithms in Bioinformatics*, S. Istrail, P. Pevzner, and M.Waterman eds., *Lecture notes in computer sciences*, **3240**, Springer, 2004, 182–193.

[2] Berger B., T. Leighton, Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete, *Computational Biology*, **5**, 1998, 27–40.

[3] Beutler T., K. Dill, A fast conformational method: A new algorithm for protein folding simulations, *Protein Sci.*, **5**, 1996, 147–153.

[4] Dill K., K. Lau, A lattice statistical mechanics model of the conformational sequence spaces of proteins, *Macromolecules*, **22**, 1989, 3986–3997.

[5] Dill K., K. M. Fiebig, H. S. Chan, Cooperativity in protein-folding kinetics, *Nat. Acad. Sci.*, USA, 1993, 1942–1946.

[6] Dorigo M., L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computing*, **1**, 1997, 53–66.

[7] Hsu H. P., V. Mehra, W. Nadler, P. Grassbergen, Growth algorithm for lattice heteropolymers at low temperature, *Chemical Physics*, **118**, 2003, 444–451.

[8] Krasnogor N., D. Pelta , P. M. Lopez, P. Mocciola, E. de la Cana, Genetic algorithms for the protein folding problem: a critical view, *Engineering of intelligent systems*, Alpaydin C. ed., ICSC Academic press, 1998, 353–360.

[9] Liang F., W. H. Wong, Evolutionary Monte Carlo for protein folding simulations, *Chemical Physics*, **115** 7, 2001, 444–451.

[10] Shmygelska A., H. H. Hoos, An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem, *BMC Bioinformatics*, **6**:30, 2005.

[11] Toma L., S. Toma, Contact interaction method: a new algorithm for protein folding simulations, *Protein Sci.*, **5**, 1996, 147–153.

[12] Unger R., J. Moult, Genetic algorithms for protein folding simulations, *Molecular Biology*, **231**, 1993, 75–81.

[13] Yue K., K. Dill, Forces of tertiary structural organization in globular proteins, *Nat. Acad. Sci.*, USA, 1995, 146–150.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Estimating time series future optima using a steepest descent methodology as a backtracker

Eleni G. Lisgara
Department of Business Administration
University of Patras
GR-265.00, Rio, Greece
Email: lisgara@upatras.gr

George S. Androulakis
Department of Business Administration
University of Patras
GR-265.00, Rio, Greece
Email: gandroul@upatras.gr

***Abstract*—Recently it was produced a backtrack technique for the efficient approximation of a time series' future optima. Such an estimation is succeeded based on a selection of sequenced points produced from the repetitive process of the continuous optima finding. Additionally, it is shown that if any time series is treated as an objective function subject to the factors affecting its future values, the use of any optimization technique finally points local optimum and therefore enables accurate prediction making. In this paper the backtrack technique is compiled with a steepest descent methodology towards optimization.**

## I. Introduction

PREDICTING future was always a rather challenging task with levels of attractiveness in the case of financial issues. Stock prices prediction, regardless its attractiveness is a very difficult task [1]. The introduction and use of time series provided the task of prediction with data history, decreasing the level of arbitrariness when executing.

Many empirical studies throuhgout literature have discussed the predictability of financial time series in respect woth the data history. In finance the daily stock prices comprise a time series, but also in meteorology, daily maximum or minimum temperatures may report one, too. Agriculture, physics, or geology, as most scientific fields interested in reporting data based on time observations, tend to produce time series reports.

Apart from the data history itself, time series has promoted into a major forecasting tool, based on statistical methodologies that use historical data to predict not future points, but the future prices, of any time series regardless their data content.

However, in the case of financial time series it is of great importance to clarify that the average investor would rather be informed about the time that the lowest and highest values will occur than the next day's– -probably ordinary—value. In mathematical means this could be translated as the time series' local minimum and maximum, respectively. The lack of applications concentrating on when the maximum or minimum would appear regardless the next point's value, led us to the obvious: *Since all known forecasting methodologies are price-oriented, it is essential to focus on a point-oriented one in order to forecast not the value of the time series, but the time that its optima will occur*.

Time series is considered as a sequence of data points arranged according to time. Let $t$ be the time intervals of time $T$; thus, the time series $Y$ is given by

$$Y = Y_t : t \in T. \tag{1}$$

On the other hand, the phenomenon represented by a time series may be also treated as a mathematical function with $m$ variables. Thus, this phenomenon may be described by an unknown but existable function $F : \mathrm{I\!R}^m \rightarrow \mathrm{I\!R}$ given by:

$$Z = F(x_1, x_2, \ldots, x_m), \tag{2}$$

where $x_1, x_2, \ldots, x_m$ are the $m$ variables' values—each depending on time—that affect the phenomenon. Obviously, the values of the time series in equation (1) are equal to those in equation (2)

$$F(x_{1,t}, x_{2,t}, \ldots, x_{m,t}) = Y_t, \qquad \forall t \in T, \tag{3}$$

where $x_{i,t}$ denotes the values of the $x_i$ in time $t$.

The unknown function $F$ may be calculated for all possible values of any variable $x_i$. However, while the phenomenon evolutes the $m$ parameters are assigned by specific arithmetic values that are smoothly modified through time. Therefore, the time series $Y$ is also the graphical representation of discrete points that lead to a curve described by a function $g(x_{1,t}, x_{2,t}, \ldots, x_{m,t})$. So, every time series is the trace of the curve $g$ along the function $F$; on this basis the time series curve could be treated as a generic function of $m$ variables.

For example, assume that the graphical representation of a random function $F$ is as shown on fig. 1, while equation's $g$ trace on it is represented by the white line. Since this curve may be represented as a generic function, its graphical representation is shown in fig. 2.

So, the time series may be represented as the trace of the curve of $g$ along the function $F$. On this basis, any time series is equivalent to a curve on the $m$-dimensional space, thus for the optima's prediction any optimization technique may be applied.

In this paper we propose a backtrack technique that allows any optimization algorithm that obtains "memory" being applied in finding future local optima. Section II includes a brief literature review of most methodologies based on which time series prediction is made. The methodology proposed is decomposed on II-C. Finally in V further research interests and applications are proposed.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

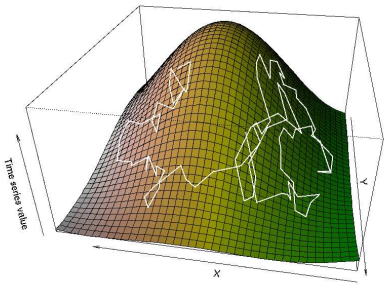


Fig. 1. The time series' general function. The white line traces the time series' real values according to its parameters values.

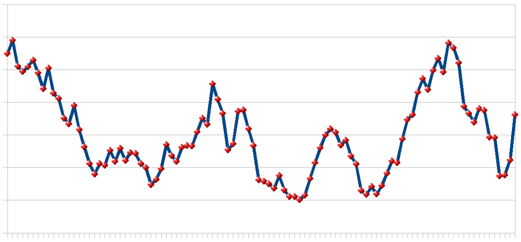


Fig. 2. The function plot of the $Y_t$ using points from the white line of 1.

## II. Time Series Forecasting

### A. Statistical Based Techniques

The widest used methodologies are based on the knowledge of *some* of the last known prices, i.e.

$$Y_t = \sum_{i=1}^{p} \Phi_i Y_(t-1) + \varepsilon_t \tag{4}$$

where $\Phi_i \in R$, and for the estimation of the new point $Y_t$ are used $p$ known points and a residual $\epsilon_t$ that satisfies

$$E[\varepsilon_t|\varepsilon_1, \ldots, \varepsilon_{t-1}] = 0 \tag{5}$$

It was H. Markowitz [2] who applied the mean-variance model on historical data and finally predicted future prices quite accurate in respect to the real ones. Based on his pioneer contribution that future points may be detected through the historical information provided by past data, and statistical assumption including means, variances and covariances, many applications in several subject areas introduced. Several bibliography on time series forecasting for finance incorporated with the probability theory. Sharpre, [3], in his pioneer work in 1970, set basis on the generalized portfolio theory of capital market, introducing the concepts of economics of risk and investment. In 1972, Merton, [4], incorporated with a set of assets by explicitly discussing the characteristics of mean-variance and efficient portfolio frontier. Further extensions added by Pang, [5], Perold ,[6], and extended the level of the parametric methods' usage in large scale selecting problems. In 1990, Best and Grauer, [7], included the general linear programming constraints and in 2000, Best and Hlouskova, [8], also included the concept of the fully and non-risky assets. The use of mean-variance efficient frontiers for the efficient assets' exchange in Jacobs et al, [9], also applied in long positions through the critical line algorithm (CLA); CLA traces out mean-variance efficient sets still including systems of linear equality or inequality constraints.

Furthermore, the distinctive introduction of the exponential smoothing model provided by Brown, [10], and Box and Jenkins, [11], arose new evidence towards predicting time series more efficiently. Such methodologies applied the so called auto-regressive integrated moving average (ARIMA) models to find the best fit of a time series on its own past values; the effectiveness of both methodologies though is a rather controversial issue according to Kuan and Lim, [12].

Most methodologies put aside the issue of the next point on which a research may focus and be interested in, and highlight on forecasting the next value. Thus, Taggart, [13], and Merton, [14], included the concept of the stock market timing in theoretical means, involving financial trends and macroeconomic policies.

### B. Artificial Intelligence Techniques

Alternative applications appeared at the late 90's, when, Lee and Jo, [15], and Edwards et al, [16], incorporated with the best time-to-market issue in terms of Artificial Intelligence to predict future stock price movements including weighted factors such as past data and market volatility.

Additionally, Pavlidis et al, [17], [18], incorporated the time series modelling and prediction through the spectrum of feed-forward neural networks as one-step local predictors applied on exchange rates. Grosan and Abraham, [1], and Chen et al, [19], incorporated results from genetic algorithms and neural network applications together, in a single multi-objective algorithm to conclude that the obtained results appear more accurate than the single use of one technique.

The common characteristic amongst these forecasting methodologies is that they neglect the aspect of *optima's* prediction in favor of that of *value*'s prediction.

### C. Optimization Techniques

*"Optimization is the act of obtaining the best result under given circumstances"* [20]. Mathematically this may be

$$\begin{aligned} \min \quad & f(X) \\ \text{s.t.} \quad & g_j(X) \leq 0, j = 1, 2, \ldots, m \\ & l_j(X) = 0, j = 1, 2, \ldots, n \end{aligned} \tag{6}$$

where $X = \{x_1, x_2, \ldots, x_n\}$ is an n-dimensional vector, $f(X)$ is called the objective function and $g_j(X)$ and $l_j(X)$ are, respectively, the inequality and equality constraints. This is the basic form of the constrained optimization problem. In the case of the lack of constraints, the problem is the unconstrained optimization one. It is common, by applying appropriate modifications, to transform any constrained problem into an unconstrained one.

A well known class of algorithms for unconstrained optimization is the Steepest Descent methods firstly introduced by Cauchy, [22],

$$x^{i+1} = x^i - \lambda_i^* \nabla F_i. \tag{7}$$

where $\lambda_i^*$ is the optimal step length along the search direction $-\nabla F_i$.

Vrahatis et al, [21], proposed a modification of the steepest descent algorithm model, called Steepest Descent with Adaptive Stepsize (SDAS-2) algorithm, based on the the Armijo's method [23].

The SDAS-2 is decomposed, using the parameters $x^0$ for the initial point, $\lambda^0$ as the randomly large initial stepsize, MIT the maximum number of iterations required and $\varepsilon$ as the predetermined desired accuracy.

**Algorithm 1** The SDAS-2 algorithm, [21]

```
Require: $\{F; x^0; \lambda^0; MIT; \varepsilon\}$
  Set $k = -1$
  while $k < MIT$ and $\|\nabla F(x^k)\| > \varepsilon$ do
    $k = k + 1$
    if $k \geq 1$ then
      Set $\Lambda^k = \frac{\|\nabla f(x^k) - \nabla F(x^{k-1})\|}{\|x^k - x^{k-1}\|}$
      if $\Lambda^k \neq 0$ then
        $\lambda = 0.5/\Lambda^k$
      else
        Set $\lambda = \lambda^0$
      end if
    else
      Set $\lambda = \lambda^0$
    end if
    Tune $\lambda$ by means of a stepsize tuning subprocedure.
    Set $x^{k+1} = x^k - \lambda \nabla F(x^k)$
  end while
  return $\{x^k; F(x^k); \nabla F(x^k)\}$
```

The following theorem—provided by Vrahatis et al, [21]—concerns the iterative scheme's convergence of the Algorithm 1:

*Theorem 1:* [21] Suppose that the objective function $F : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and bounded below on $\mathbb{R}^m$ and assume that $\nabla F$ is Lipschitz continuous on $\mathbb{R}^m$. Then, given any $x^0 \in \mathbb{R}^m$, for any sequence $\{x^k\}_{k=0}^{\infty}$, generated by Algorithm 1, satisfying the Wolfe's conditions [24], [25], [26] implies that $\lim_{k\to\infty} \nabla F(x^k) = 0$.

## III. Decomposition of the Backtrack Procedure

As already mentioned in I the time series $Y$ is also the trace of a curve along function $F$ of $m$ variables. Thus, the problem of time series' optima search is equivalent to the constrained optimization problem (equation 6).

Using the notation of equation (6) lots of the unconstrained minimization methods are iterative in nature and hence they start from an initial random solution, $X^0 \in \mathbb{R}^m$ and proceed towards the minimum point in a sequential manner, $x^1, x^2, \ldots x^{\min}$. According to the time series notation in equation (1), point $t_i$ is denoted by $t_i = (x_1^i, x_2^i, \ldots, x_m^i)$.

Without lack of generality it is supposed that the prediction of future maximum is in question; the procedure is decomposed in two stages:

1) **Backward search for minimum** When applying any optimization technique, it is concluded that most of them in order to generate the new point use prior knowledge collected from the process data including points, function values, gradient values, matrix approximations etc. In our case, let $t_n$ be the last known point of time series, and by using the SDAS-2 algorithm the next estimation of local minimum is calculated when applying Algorithm 1. Obviously, in order to calculate an approximation of the $\nabla F$ we have used finite differences. Therefore, by applying the SDAS-2, the sequence of points that leads to a local minimum $[t_n, t_{k_1}, t_{k_2}, \ldots, t_{k_{m-2}}, t_{\min}]$, where $n > k_1 > k_2 > \cdots > k_{m-2} > \min$ is estimated.
2) **Forward search for minimum** When this "sequence" of points is viewed as a forward process, it appears as a sequence that starts from the minimum past point $t_{\min}$, crisscrosses the last known point $t_n$ and probably leads to a maximum future one $t_{\max}$. In this forward process the stepsize $\Lambda_i^*$ and the sequence of points are known from the previous stage 1. The constructed sequence of points provides us with all the information needed to proceed in estimating a future maximum. So, by applying one step of Algorithm 1 from the initial point $t_n$ an approximation of $t_{\max}$ is obtained.

Thus, the previously described process gives the following backtrack algorithm for the estimation of future local maximum; note that $\varepsilon$ is a very small positive number that reassures desired accuracy.

**Algorithm 2** The backtrack algorithm

```
Require: $\{t_n, \varepsilon\}$
  repeat
    Run stage 1 to compute a sequence of points $t_n$, $t_{k_1}$, $t_{k_2}$, $\ldots$, $t_{k_{m-2}}$, $t_{\min}$ that leads to "past" local minimum.
    Calculate "future" point $t_{n+k}$ by applying stage 2 using points $t_{k_{m-2}}$, $\ldots$, $t_{k_2}$, $t_{k_1}$, $t_n$.
    Set $t_n = t_{n+k}$
  until $Y'_{t_n} < \varepsilon$
  Set $t_{\max} = t_{n+k}$.
  return $t_{max}$
```

Note that the procedure described in Algorithm (2) is equivalent to the optimization problem given by equation (6). Therefore, since all assumptions of Theorem 1 are satisfied, the Algorithm 2 converges to an optimum future point.

## IV. Numerical Applications

To investigate the proposed method's reliability, the method was applied on two different time series samples; one fi-

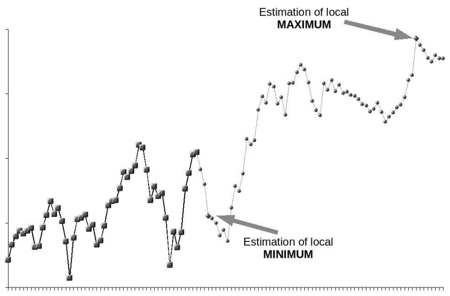


Fig. 3. An application of the backtrack algorithm on Athens Stock Market general index on April 14, 1999

nancial stock market data and another with meteorological ones.

## A. Athens' Stock Market

The proposed application is tested on the daily closing prices of the Athens' Stock Market. The data consists of the daily closing prices of 18 years—from 1985 until 2002. In Figure 3 is presented an application of the backtrack algorithm towards predicting Athens' Stock Market general index for the randomly chosen date of April 14, 1999. The last 50 known values of general index are used; i.e. in the case of April 14, 1999, the 50 last known indexes are from January 29, 1999 until April 14, 1999. These points are represented on Figure 3 with the square symbol. In Figure 3, again, the gray circles stand for the index's actual values index for the exchanging period from April 15, 1999 till July 14, 1999. Future values are connected together with the discontinuous line. The points depicted from the backtrack algorithm are symbolized with the rhomb symbol.

In Figure 4 is presented an application of the SDAS-2 backtrack algorithm towards predicting Athens' Stock Market general index for the randomly chosen date of July 13, 1998; then Algorithm 2 is applied to approximate future local minima and maxima. The last 50 known values of the general index are used; i.e. in the case of July 13, 1998, the 50 last known indexes are from May 4, 1998 until July 13, 1998. These points are represented on Figure 4 with the square symbol. In Figure 4, again, the gray circles stand for the index's actual values for the exchanging period from July 14, 1998 till September 16, 1998. Future values are connected together with the discontinuous line. The points depicted from the SDAS backtrack algorithm are symbolized with the rhomb symbol.

To investigate the SDAS backtrack algorithm's reliability, the algorithm was applied on 1000 randomly chosen prices to approximate the dates $t_{min}$ and $t_{max}$ that produce local minima and local maxima, respectively. Since we assume that the investor seeks for assets that are relatively cheap to buy and their price increase, the table only includes the cases that the local minima emerges before the local maxima.

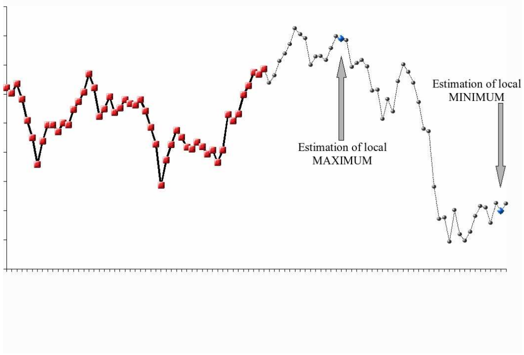


Fig. 4. An application of the SDAS backtrack algorithm on Athens Stock Market general index on July 13, 1998

TABLE I
STATISTICAL INDEXES APPLIED ON RANDOMLY CHOSEN ASSETS

| Quantity | Days | Return |
|---|---|---|
| **Min** | 1.00 | -0.2320 |
| $Q_1$ | 4.00 | 0.0020 |
| **Median** | 9.00 | 0.0520 |
| **Mean** | 15.54 | 0.0662 |
| $Q_3$ | 21.25 | 0.1162 |
| **Max** | 55.00 | 0.4020 |
| **s.d.** | 15.65 | 0.1343 |

The difference between the estimations for local minima and local maxima is extracted in terms of time, that is days of market activity. Furthermore, each asset's return was calculated based on the function

$$R = \frac{Y_{t_{\max}} - Y_{t_{\min}}}{Y_{t_{\min}}} \tag{8}$$

In Table I are briefly reported the statistical indexes that characterize the variables "days" and "return" obtained from equation 8. Specifically, the raw "days" includes indexes regarding the difference in market days between local minima and local optima, $t_{\max} - t_{\min}$; likewise, raw named "return" illustrates the indexes that result from variable $R$ estimated from equation (8). The abbreviations used as columns are referred to the local minimum, first quarter $Q_1$, mean value, median, third quarter $Q_3$ and standard deviation.

It is therefore, observed that the mean between the predicted local minima and local maxima is 15 days approximately. During this period the mean return is about $6.6\%$.

## B. Athens Temperature

As widely known, the time series derived from meteorological data appear to have several local optima due to weather's dependence from sensitive factors. We have constructed a time series based on the data collected from the National Technical University of Athens' meteorological station, that includes information about the temperature for every 10 minutes from February 14, 1999 until today.

If the time series' graph is closely observed, for instance for the Christmas Day of 2006 (Fig. 5), it is obvious that

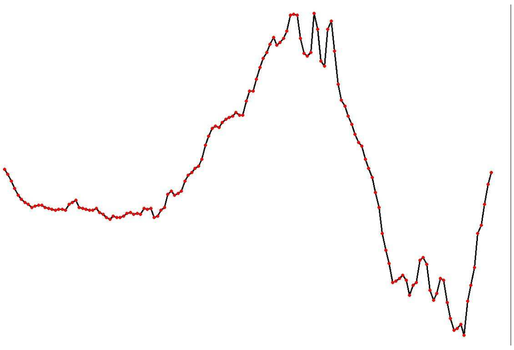


Fig. 5. Athens' temprature during Christmas Day of 2006.

TABLE II
STATISTICAL INDEXES APPLIED ON RANDOMLY CHOSEN ASCENDING TEMPERATURES

| Quantity | Forecasted Max | Forecasted Min |
|---|---|---|
| **Min** | -1.16 | -8.93 |
| $Q_1$ | -0.20 | -0.21 |
| **Median** | 0.03 | 0.01 |
| **Mean** | 0.20 | -0.08 |
| $Q_3$ | 0.24 | 0.28 |
| **Max** | 4.64 | 3.62 |
| **s.d.** | 0.99 | 1.39 |

there is a high number of optima. Furthermore, there are areas of continuous changes from minima or maxima, especially around the global maxima—occurred in 15.00—and the global minima—occurred in 22.30.

The proposed algorithm was tested on the time series data for the year 2006. In order to avoid the effects derived from the continuous changes we have randomly chosen points that appear to follow an ascent or a descent stream. It is supposed that a time series follows an ascent stream when its $m$ last known points are in an ascent order, too. Furthermore, when the time series appeared following an ascent stream, the backtrack algorithm was applied for the detection of the past known minima; the SDAS-2 algorithm was applied only for the prediction of the future local maxima. Samewise, if the time series appeared following a descent stream the backtrack algorithm was applied for the detection of the past maxima, and the SDAS-2 for the prediction of the local minima.

The results obtained from ascending temperature data are shown on Table II while those obtained from descending temperature are shown on Table III. In order to appraise the statement that for the ascending stream case the level of successfulness in predicting local maxima is greater than this for the local minima and vice-verca, table IV was constructed, it contains the successfulness level for each case.

The values shown on both "Forecasted Max" and "Forecasted Min" columns conclude from the difference between the value of the predicted local optimum point and the value of the initial known one.

TABLE III
STATISTICAL INDEXES APPLIED ON RANDOMLY CHOSEN DESCENDING TEMPERATURES

| Quantity | Forecasted Max | Forecasted Min |
|---|---|---|
| **Min** | -2.24 | -5.36 |
| $Q_1$ | -0.43 | -0.38 |
| **Median** | -0.11 | -0.07 |
| **Mean** | 0.16 | -0.37 |
| $Q_3$ | 0.34 | 0.10 |
| **Max** | 4.00 | 0.56 |
| **s.d.** | 1.24 | 1.12 |

TABLE IV
LEVELS OF SUCCESSFULNESS IN PREDICTING MAXIMUM AND MINIMUM IN CO-RELATION WITH THE TIME SERIES' STREAM

| Direction | Min | Max |
|---|---|---|
| Ascending | 47% | 52% |
| Descending | 55% | 38% |

The column "Direction" illustrates the timeseries' direction, in means of an "ascending" or a "descending" stream. In column "Min" is shown the level of successfulness in predicting local minima; same wise, the column "Max" demonstrated the successfulness level when predicting local maxima. It is observed that the local maxima's prediction while the timeseries follows an ascending stream is quite better than this of the local minima, and vice-versa. However, such behavior is rather expected due to the usage of the SDAS-2 algorithm that belongs to the category of the steepest descent algorithms. According to [27] this category of techniques tends to converge to the closest optima.

## V. Conclusions and Further Research Interests

This paper is structured based on the rationalization that the financial time series may be treated as a function subjected to those that represent all different factors affecting its values during time, thus we incorporated with optimization techniques instead of statistical ones. Obviously, such a rationalization provides strong enough evidence towards applying any optimization technique. Due to this, the proposed backtrack technique was applied using the SDAS-2 algorithm incorporated in [21]. The results obtained provide strong evidence regarding predictions' accuracy. Further, it is observed that the local maxima's prediction—while the timeseries follows an ascending stream—is quite better than this of the local minima, and vice-versa.

The SDAS-2 algorithm was applied since

(a) uses prior knowledge (calculates an approximation of Lipschitz constant using all sequence points), and
(b) as proposed by [27] the steepest descent methods are the most reliable as far as it concerns the convergence to the closest optimum; both characteristics could applied in portfolio optimization problems.

At this point, three directions could be proposed based on the backtrack technique: (a) applications of the backtrack technique in other minimization algorithms that use information collected from sequence points, such as quasi-Newton methods, conjugate gradient, etc, (b) using these techniques towards approximate a future minimum and a future maximum, as well, of an individual asset, so as to formulate an asset management technique for its behavior in an asset portfolio and (c) using backtrack technique with a multi-objective optimization method for managing a given portfolio.

Further research could be focused on these directions, due to the interesting results the technique provides; our research concentrates towards these directions, as well.

## REFERENCES

[1] C. Grosan and A. Abraham, *Stock Market Modeling Using Genetic Programming Ensembles*. Berlin/Heidelberg: Springer, 2006.

[2] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, pp. 77–91, 1952.

[3] W. Sharpe, *Portfolio Theory and Capital Markets*. McGraw–Hill, New York, 1970.

[4] R. Merton, "An analytic derivation of the efficient frontier," *Journal of Finance and Quantitative Analysis*, vol. 9, pp. 1851–1872, 1972.

[5] J. Pang, "A new efficient algorithm for a class of portfolio selection problems," *Operational Research*, vol. 28, pp. 754–767, 1980.

[6] A. Perold, "Large-scale portfolio optimization," *Management Science*, vol. 30, pp. 1143–1160, 1984.

[7] M. Best and R. Grauer, "The efficient set mathematics when mean-variance problems are subject to general linear constrains," *Journal of Economics and Business*, vol. 42, pp. 105–120, 1990.

[8] M. Best and J. Hlouskova, "The efficient frontier for bounded assets," *Mathematical Methods of Operations Research*, vol. 52, pp. 195–212, 2000.

[9] B. Jacobs, K. Levy, and H. Markowitz, "Portfolio optimization with factors, scenarios, and realistic short positions," *Operations Research*, vol. 53, pp. 586–599, 2005.

[10] R. G. Brown, *Statistical Forecasting for Inventory Control*. McGraw-Hill, New York, 1959.

[11] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. 2nd Edition, Holden–Day, San Francisco, 1976.

[12] C. M. Kuan and T. Lim, "Forecasting exchange rates using feedforward and recurrent neural networks," *Journal of Applied Econometrics*, vol. 10, pp. 347–364, 1994.

[13] R. A. Jr Taggart, "A model of corporate financing decisions," *The Journal of Finance*, vol. 32, pp. 1467–1484, 1977.

[14] R. C. Merton, "On market timing and investment performance. i. an equilibrium theory of value for market forecasts," *The Journal of Business*, vol. 54, pp. 363–406, 1981.

[15] K. Lee and G. Jo, "Expert system for predicting stock market timing using a candlestick chart," *Expert Systems with Applications*, vol. 16, pp. 357–364, 1999.

[16] R. D. Edwards, J. Magee, and W. H. Bassetti, *Technical Analysis of Stock Trends*. AMACOM, Division of the American Management Associations, 2001.

[17] N. G. Pavlidis, V. P. Plagianakos, D. K. Tasoulis, and M. N. Vrahatis, "Financial forecasting through unsupervised clustering and neural networks," *Operational Research: An International Journal*, vol. 6, no. 2, pp. 103–127, 2006.

[18] N. G. Pavlidis, D. K. Tasoulis, V. P. Plagianakos, and M. N. Vrahatis, "Computational intelligence methods for financial time series modeling," *International Journal of Bifurcation and Chaos*, vol. 16, no. 7, pp. 2053 – 2062, 2006.

[19] Y. Chen, L. Peng, and A. Abraham, "Stock index modeling using hierarchical radial basis function networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4253 LNAI - III, pp. 398–405, 2006.

[20] S. S. Rao, *Optimization, Theory and Applications*. New Delhi: Wiley Eastern Limited, 1984.

[21] M. N. Vrahatis, G. S. Androulakis, J. N. Lambrinos, and G. D. Magoulas, "A class of gradient unconstrained minimization algorithms with adaptive stepsize," *Journal of Computational and Applied Mathematics*, vol. 114, pp. 367 – 386, 2000.

[22] A. Cauchy, "Méthode générale pour la résolution des systémes d'équations simultanées," *Comp. Rend. Acad. Sci. Paris*, vol. 25, pp. 536–538, 1847.

[23] L. Armijo, "Minimization of function having Lipschitz continous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, pp. 1–3, 1966.

[24] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Review*, vol. 11, pp. 226–235, 1969.

[25] ——, "Convergence conditions for ascent methods II: some corrections," *SIAM Review*, vol. 13, pp. 185–188, 1971.

[26] J. Nocedal, "Theory of algorithms for unconstrained optimization," in *Acta Numerica 1991*, A. Iserles, Ed. Cambridge University Press, Cambridge, 1991, pp. 199–242.

[27] G. S. Androulakis and M. N. Vrahatis, "OPTAC: A portable software package for analyzing and comparing optimization methods by visualization," *Journal of Computational and Applied Mathematics*, vol. 72, pp. 41–62, 1996.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 899–906

ISBN 978-83-60810-14-9
ISSN 1896-7094

# On a class of periodic scheduling problems: models, lower bounds and heuristics

Philippe Michelon
and Dominique Quadri
Université d'Avignon
Laboratoire d'Informatique d'Avignon
F-84911 Avignon Cedex 9, France
Email: {philippe.michelon,
dominique.quadri}@univ-avignon.fr

Marcos Negreiros
Universidade Estadual do Ceará
Departamento de Estatística e Computação
Av. Paranjana, 1700 Fortaleza, Brasil
Email: negreiro@uece.br

***Abstract*—We study in this paper a generalization of the basic strictly periodic scheduling problem where two positive integer constants, associated to each task, are introduced such that to replace the usual strict period. This problem is motivated by the combat of the dengue which is one of the major tropical disease. We discuss general properties and propose two integer mathematical models of the problem considered which are compared theoretically. We also suggest a lower bound which is derived from the structure of the problem. It appears to be quickly obtained and of good quality. Three greedy algorithms are proposed to provide feasible solutions which are compared with the optimum (when it can be obtained by the use of ILOG-Cplex10.0). It is shown that for special instances greedy algorithms are optimal.**

## I. Introduction

DENGUE is a flu-like viral disease spread by the bite of infected mosquitoes and occurs in most tropical areas of the world. One of a severe complication of this disease is dengue hemorrhagic fever which is often fatal. Unfortunately, there is no specific treatment for it. Consequently at the present time the only way of preventing or of fighting the dengue is to eliminate the vector mosquitoes which are located in breeding sites. More specifically equipped vehicles are sent in each infested and detected areas to pulverize insecticide. In practice those sites are divided in sub-areas which are computed so as to exactly obtain one working day for each vehicle (formally each task takes a duration of unite time). Unfortunately, the pulverized product only kills mosquitoes but leaves the larves in life. Indeed, it takes between 7 and 9 days for a larve to become an adult mosquito. Therefore, each sub-area has to be treated repetitively with a delay of at least 7 and at most 9 days between two pulverizations to achieve the best efficiency. We refer to [22] for more details on the logistic aspect of prevention and combat of the dengue. More formally, we consider the problem of minimizing the number of vehicles required to make periodic single destination equipped vehicle to a set of infested sub-areas. We therefore face to a periodic scheduling problem (which is in this basic form NP-hard [10]) where the periodicity is not strict but represented by both a minimum and a maximum delay (respectively, in this special case 7 and 9). We show in this paper that if all the maximum periods are equals for all tasks then the problem relative to our application of dengue prevention can be solved in polynomial time. We will then also study the more general case where all the maximum periods are different.

We therefore extend our work to a generalization of a strict periodic scheduling problem which is basically concerned with processing, on a set of identical machines (or identical unitary resources), periodic tasks or activities over an infinite horizon. Each activity $i$ is characterized by a duration $d_i$. We are actually concerned with the following generalization of the strict periodicity requirement: two positive integer numbers $\underline{F_i}$ and $\overline{F_i}$ are associated with task $i$ and correspond, respectively, to a minimum and a maximum delay for the repetition of activity $i$. Thus, if the $k^{th}$ execution of $i$ has been scheduled on time $t_{i,k}$ then $t_{i,k+1}$ must belong to $[t_{i,k}+d_i+\underline{F}_i, t_{i,k}+d_i+\overline{F}_i]$. We also consider a finite time horizon, all the task durations equals to 1 (i.e. $d_i = 1 \quad \forall i$) and unary resources. Our aim is to minimize the number of resources (or machines or vehicles) so as to execute all the activities.

In this paper, we examine the structure of this general problem (denoted by $GSPS$), giving some properties. We propose two integer linear formulations which we name "weak and strong" formulations for $GSPS$. We compare theoretically those formulations that asserts the named of each model. From the strong formulation we derived a lower bound denoted by $\lceil Z[\overline{GSPS}]\rceil$. We then suggest a second lower bound of $GSPS$ easier to be computed and of good quality which appears to be optimal for special cases, including when all the maximum delay are equals. Finally, we present three greedy algorithms which provide feasible solutions. Those upper bounds are evaluated and compared with both the optimal value of the integer linear model when it can be obtained in a competitive CPU time given by CPLEX10.0. and the lower bounds we proposed.

The paper is organized as follows. In Section II, we formally define the problem $GSPS$ and give necessary notations. Section III describes the relevant literature. We establish in Section IV some properties relative to the considered scheduling problem and provide a trivial lower bound. Section V is dedicated to the formulation of $GSPS$ by two integer linear programs.

In Section VI we propose three greedy algorithms so as to obtain good feasible solutions. The computational results are reported in Section VII. In Section VIII we summarize the main results of this paper and we point out some directions for future research.

## II. Problem definition

We consider in this paper a problem denoted by $GSPS$ which is a generalization of both the basic strictly periodic scheduling problem [10] and the problem derived from the combat of the dengue described in the introduction.

Formally, it can be stated as follows. We consider $J$ types of activities and associate at each type $j$ ($j = 1, .., J$) the following parameters:

- $n_j$, the number of activities of type $j$.
- $\underline{F}_j$ the minimum delay between two executions of an activity of type $j$.
- $\overline{F}_j$ the maximum delay between two executions of an activity of type $j$.

Each activity of each type requires an unary resource (at each time it is processed) and has a duration of a unit time. The resources are identical and renewable (i.e. right after having processed a task, the resource is available for another task).

The objective of the problem is to find a feasible schedule, with respect to the minimum and maximum delays between two executions of the same activity, over a time horizon $H$ while minimizing the number of used resources. The activities of type $j$ are required to be executed at least once in the first $\overline{F}_j$ units of time.

The following proposition establishes the complexity of the problem we study.

*Proposition 1:* The problem $GSPS$ considered here is NP-Hard.

*Proof:* If $\underline{F}_j = \overline{F}_j$, $\forall j = 1...J$ then $GSPS$ corresponds to the strictly periodic problem which has been shown as being NP-Hard (see [10] and [24]). Consequently, the strictly periodic scheduling problem is a special case of $GSPS$. Since it is NP-hard then $GSPS$ remains NP-hard. ■

To get a clearer reading, let us provide an example of a feasible solution (which actually happens to be optimal) represented by Figure 1. This solution uses 4 units of resource or machines (or vehicles if we refer the dengue application) over an horizon of 20 units of time, for the problem corresponding to the following data:

| type | n | $\underline{F}$ | $\overline{F}$ | activities |
|---|---|---|---|---|
| 1 | 3 | 0 | 2 | 1,2,3 |
| 2 | 2 | 2 | 3 | 4,5 |
| 3 | 1 | 2 | 4 | 6 |
| 4 | 2 | 3 | 4 | 7,8 |
| 5 | 1 | 6 | 6 | 9 |

where the first column corresponds to the type identifier, the second column reports the number of activities for the relative type, the third and fourth columns give respectively the minimum and maximum delay between 2 executions of an activity of this type and the last ones indicates which are the activities of this type.

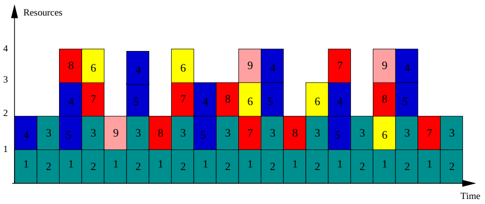


Fig. 1. A feasible solution

## III. Literature review

We review in this section the literature relative to periodic scheduling problems. More specifically, we begin by presented the state of art concerning the strictly periodic scheduling (and/or routing) problem. We note that the major attention with regard to this problem has been addressed to heuristics methods. We then pursue the literature concerning variant and generalization of the previous problem. We finally briefly describe other applications of periodic scheduling problems.

In a basic version of strictly periodic scheduling problem, all durations are equals to 1 and a period $T_i$ is also associated with each of the task so that if task $i$ ($\forall i = 1, ..., n$) is scheduled on time $t$ then it must also be strictly scheduled on times $t + d_i + T_i$, $t + 2d_i + 2T_i$ and so on. The problem consists then in minimizing the number of machines to process periodically the tasks. This basic problem has been shown as being NP-hard ([10], [24]) although it has also been shown that it is sufficient to compute a schedule over a time horizon equal to the lowest common multiplier of the $T_i$ (since the latest can be duplicated for larger time horizons [10]).

Generalizations of this basic periodic problem have then been introduced. Jan Korst's Ph.D. thesis [19] considers a periodic scheduling problem very similar but with general integral execution times. In Park and Yun [23] and Gaudioso et. al. [11], unitary duration times are considered with task $i$ requires $w_i$ units of resources, while being executed. Both problems are identical and do correspond with the basic problem when all $d_i$ and $w_i$ values equal one. A large integer program was then introduced by [23] and [12] to solve the resulting load minimization problem, however Park and Yun [23] also describe a method for decomposing the initial problem into smaller subproblems. They divide the set of activities into sets $N_1, N_2, \cdots, N_d$, where for any $i \in N_q$ and any $j \in N_r$, $p_i$ and $p_j$ (periods) are relatively prime. They then use integer programming to solve each resulting subproblem, minimizing the number of resources required for each subset of customer. If $K_q$ represents the minimum number of resources required for customer set $N_q$, Park and Yun [23] showed that the minimum number of vehicles required to service the customers is exactly $K_1 + K_2 + \cdots + K_d$ since the decomposition does not increase the number of vehicles required. Finally, Gaudioso et al. [12] also present a branching heuristic with several possible branching rules, where the greedy algorithm presented in [10] corresponds to one of the choice of these rules. Another generalization has been considered in [5], also in the strictly periodic case, by considering non unitary resources and non unary demands associated with every activities have a duration of 1 time unit. The concept of tree scheduling is then introduced, which is a methodology for developing perfectly periodic schedules based on hierarchical round-robin, where the hierarchy is represented by trees. Some optimal (exponential time) and efficient heuristic algorithms which generate schedule trees are presented.

Some authors relax the strict periodicity requirement ([14]): rather than exactly scheduling activity $i$ on times $t + d_i + T_i$, $t + 2d_i + 2T_i$, ..., it is allowed to be scheduled on time slots centered in $t + d_i + T_i$, $t + 2d_i + 2T_i$, ... Namely, the second execution of task $i$ must occur in $[t+d_i+T_i-a_i, t+d_i+T_i+b_i]$, the third in $[t + 2d_i + 2T_i - a_i, t + 2d_i + 2T_i + b_i]$, etc. where $a_i$ and $b_i$ are non negative real numbers associated with $i$. Baruah et al. [8] introduce the notion of Proportionate fairness (PFair) Scheduling. PFair scheduling differs from more conventional real-time scheduling approaches in that tasks are explicitly required to execute at steady rates and have been deeply studied [9], [7] and [3]. This (multiple-resource) periodic scheduling problem was first addressed by Liu[21]. Baruah et al. [8] showed that PFair scheduling can be solved in polynomial time.

Other closer works are found also in the periodic routing problems (PRP). In [10], the authors consider the problem of minimizing the number of vehicles required to make strictly periodic, single destination deliveries to a set of customers, under the initial assumption that each delivery requires the use of a vehicle for a full day. A greedy algorithm that is optimal in some special cases is proposed. A variant of this problem is also considered when the restriction of a full day for each delivery is relaxed. The same relaxation proposed in this article is also considered in [11], however, the authors address conjointly the routing problem for each day, which makes the problem much more difficult to solve due to the routing component and its interactions with the periodic scheduling problem.

Early work on periodic schedules was also motivated by Teletext Systems [2], [13] and [1], maintenance problem [4], [25] and [20] and broadcast disks [6], [18], [16] and [17]. This latter issue gained a lot of attention recently since they are used to model backbone communication in wireless systems, Teletext systems and efficient web caching in satellite systems. Finally in [15], the authors address how to sequence the movements of robots in order to minimize the number of robots required to complete a series of tasks over a fixed time horizon. In this case, the periodicity comes from the sequence of each robot's movement rather than from the jobs being processed.

## IV. Some properties and a trivial lower bound

We dedicated this section to the study of the structure of the problem $GSPS$. We thus first exhibit some observations and properties. We then provide a simple way to compute a lower bound of good quality for $GSPS$ derived from the structure of the problem which becomes to be optimal when all the maximum periods are equals.

*Observation 1:* As proved in [10] if the problem is a strictly periodic scheduling then to obtain a solution over an infinite time horizon, it is necessary only to find a solution over a time horizon equal to the lowest common multiplier of the periods. On the opposite, if we consider our problem $GSPS$, it is not feasible to duplicate over an infinite horizon a schedule computed over a horizon equal to the lowest common multiplier of the maximum period. Indeed, since the period is not strict any more then the previous property can not be established.

*Observation 2:* The decomposition procedure of the basic periodic scheduling problem, into sub-problems easier to be solved, suggested by Park and Yun [23] can not be extended in our context because the periodicity is not any more strict.

The two previous observations are straightforwardly validated by counter-examples, which are easy to be found.

In spite of the fact that the more general periodicity seems to be inconvenient for using the lowest common multiplier of the periods to establish properties, this notion still plays a key role in the computation of lower and upper bounds for our problem $GSPS$. Indeed, we present now a simple technique to compute a lower bound of $GSPS$ which becomes easier if the horizon time $H$ is greater than the common multiplier of the maximum delays $\overline{F_j}$, $\forall j = 1...J$. The upper bounds will be provided by greedy algorithms in Section VI.

A simple way to under-evaluate the number of needed resource units is to find how much time each activity has to be processed. For this purpose, let us consider a time $t$ ($\leq H$). Between times 1 and $t$, an activity of type $j$ will be processed at least $\lfloor \frac{t}{\overline{F_j}} \rfloor$ and therefore, between times 1 and $t$, we have to process at least $\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F_j}} \rfloor$ tasks, hence an average of:

$$\frac{\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F_j}} \rfloor}{t} \qquad (1)$$

per time unit. Since the average of a set of values is smaller than the maximum, the smallest integer greater than quantity (1) (that is the ceil of (1)) is a lower bound to the number of resources needed to schedule all the activities between times 1 and $t$. In order to get the best (over the set of those lower bounds) lower bound for scheduling all the activities of all the types over the time horizon, we then have to compute;

$$TLB = \max \left\{ \left\lceil \frac{\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F_j}} \rfloor}{t} \right\rceil / \quad t = 1, ..., H \right\} \qquad (2)$$

The above quantity will be referred as the "Trivial Lower Bound" (TLB).

The following proposition establishes the role of the lowest common multiplier of the $\overline{F}_j$ $(j = 1, ..., J)$ in the computation of the (TLB).

*Proposition 2:* If the time horizon $H$ is greater or equal than the lowest common multiplier of the $\overline{F}_j$ $(j = 1, ..., J)$, then the (TLB) is actually equal to:

$$TLB = \left\lceil \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} \right\rceil \tag{3}$$

*Proof:* Note first that $TLB \leq \left\lceil \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} \right\rceil$ : since $\lfloor \frac{t}{\overline{F}_j} \rfloor \leq \frac{t}{\overline{F}_j}$ , we obviously have:

$$\frac{\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F}_j} \rfloor}{t} \leq \frac{\sum_{j=1}^{J} n_j \frac{t}{\overline{F}_j}}{t} = \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} \tag{4}$$

It then follows that:

$$TLB = \max \left\{ \left\lceil \frac{\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F}_j} \rfloor}{t} \right\rceil \Big/ \quad t = 1, ..., H \right\} \leq \left\lceil \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} \right\rceil \tag{5}$$

Let now $F$ denote the lowest common multiplier of the $\overline{F}_j$ $(j = 1, ..., J)$. If $H \geq F$, we can consider inequality (4) for $t = F$. Since, $\frac{F}{\overline{F}_j}$ is integer, the inequality $\lfloor \frac{t}{\overline{F}_j} \rfloor \leq \frac{t}{\overline{F}_j}$ becomes an equality and, thus,

$$TLB = \max \left\{ \left\lceil \frac{\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F}_j} \rfloor}{t} \right\rceil \Big/ \quad t = 1, ..., H \right\} \geq \left\lceil \frac{\sum_{j=1}^{J} n_j \frac{F}{\overline{F}_j}}{F} \right\rceil \tag{6}$$

Since,

$$\left\lceil \frac{\sum_{j=1}^{J} n_j \frac{F}{\overline{F}_j}}{F} \right\rceil = \left\lceil \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} \right\rceil \tag{7}$$

we have,

$$TLB = \max \left\{ \left\lceil \frac{\sum_{j=1}^{J} n_j \lfloor \frac{t}{\overline{F}_j} \rfloor}{t} \right\rceil \Big/ \quad t = 1, ..., H \right\} = \left\lceil \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} \right\rceil \tag{8}$$

■

We show now, as mentioned in the introduction, that the application of the combat of the dengue, which represents a particular case of $GSPS$ can be solved in polynomial time.

*Proposition 3:* If all the types have the same maximum delay $\overline{F}$ (i.e. $\overline{F}_j = \overline{F}$ $\forall j = 1, ..., J$) then the problem $GSPS$ can be solved to optimality in $\lfloor \frac{H}{\overline{F}} \rfloor \sum_{j=1}^{J} n_j$ operations, that is in polynomial time with respect to the total number of tasks to be scheduled. In addition, the optimal value is exactly equal to the value provided by the Trivial Lower Bound.

*Proof:* With no loss of generality, we can assume that $H \geq \overline{F}$ (otherwise, it is not necessary to execute the activities). Let us consider a particular schedule that we are going to show as being feasible and which provides a objective function value equal to (TLB). As a consequence, this schedule will be shown as being optimal. This particular schedule is builded by the following greedy method:

- If $\sum_{j=1}^{J} n_j \leq \overline{F}$ then the problem is rather easy to solve : assign activity 1 on times 1, $1 + \overline{F}$, $1 + 2\overline{F}$ and so on; activity 2 on times 2, $2 + \overline{F}$, $2 + 2\overline{F}$,... etc ...
  It is then direct to observe that this procedure will use only one unit resource and that each activity will be executed with a strict period of $\overline{F}$ units of time. Thus, the produced schedule is feasible and optimal since no less than 1 resource unit can be used.
- Otherwise, i.e. if $\sum_{j=1}^{J} n_j > \overline{F}$, then we take the first $\overline{F}$ activities and assign activity 1 on times 1, $1 + \overline{F}$, $1 + 2\overline{F}$ and so on, activity 2 on instants 2, $2 + \overline{F}$, $2 + 2\overline{F}$,... etc ...
  Consequently, each of those activities are executed according to a strict period of $\overline{F}$ units of time and on one unit of resource.
  Repeat the same process for the next $\overline{F}$ activities and so on until that the number of remaining activities is no greater than $\overline{F}$. Then apply the same procedure as above. Once again, each of the activities are executed exactly $\overline{F}$ units of time (hence, the schedule is feasible) and after an easy calculation it can be straightforwardly established that this schedule requires $\left\lceil \frac{\sum_{j=1}^{J} n_j}{\overline{F}} \right\rceil$ units of resource. Since this quantity is exactly equal to the Trivial Lower Bound, the schedule is actually optimal.

■

## V. INTEGER LINEAR MODELS

In this Section, we first introduce two integer mathematical programs modelling $GSPS$ for the general periodic scheduling problem. Then, a theoretically comparison of the models is presented *via* corollary 1. Consequently, we will refer to a "weak formulation" and to a "strong formulation". We begin by establishing the so-called "weak formulation".

First are introduced some necessaries notations. Let $J(i) \in \{1, 2, ..., J\}$ be the type of activity $i$ and $n = \sum_{j=1}^{J} n_j$ the total number of activities to process.

Let us now define the following decision variables:

$$x_{it} \ = \begin{cases} 1 & \text{if task } i \text{ is processed on time } t \\ 0 & \text{otherwise} \end{cases}$$

where $i = 1, ..., n$ and $t = 1, ..., H$.

$R \ = $ the number of unit resources that we want to minimize

Let also $J(i) \in \{1, 2, ..., J\}$ be the type of activity $i$ and $n = \sum_{j=1}^{J} n_j$ the total number of activities to process. Then, the "weak formulation" can be stated as follows:

$$GSPS \begin{cases} \min \ R \\ s.t. \left| \begin{array}{l} (9), (10) \ or \ (12), (11) \\ x_{it} \in \{0, 1\}, \ \forall i = 1, ..., n, \ \forall t = 1, ..., H \\ R \geq 0 \end{array} \right. \end{cases}$$

where contraints (9), (10), (11) and (12) are respectively defined as follows.

At each time $t$, the number of used resources is greater or equal than the number of scheduled activities. Consequently we have constraint (9):

$$R \geq \sum_{i=1}^{n} x_{it} \quad \forall t = 1, ..., H \tag{9}$$

If activity $i$ is scheduled on instant $t$, then it must also be scheduled between instants $t + 1 + \underline{F}_{j(i)}$ and $\overline{F}_{j(i)}$. It follows constraint (10):

$$\sum_{l=t+1+\underline{F}_{j(i)}}^{t+\overline{F}_{j(i)}} x_{il} \geq x_{it} \qquad \forall i = 1, ..., n \ \ \forall t = 1, ..., H - \overline{F}_{j(i)} \tag{10}$$

Finally, each activity can be scheduled at most once on each period of $\underline{F}_{j(i)}$ consecutive days. We are then face to constraint (11):

$$\sum_{l=t}^{t+\underline{F}_{j(i)}} x_{il} \leq 1 \qquad \forall i = 1, ..., n \ \ \forall t = 1, ..., H - \underline{F}_{j(i)} \tag{11}$$

The "strong formulation" we proposed is obtained by substituting constraint (10) by:

$$\sum_{l=t+}^{t+\overline{F}_{j(i)}} x_{il} \geq 1 \qquad \forall i = 1, ..., n \ \ \forall t = 1, ..., H - \overline{F}_{j(i)} \tag{12}$$

Actually, this constraint states that, on each period of $\overline{F}_{j(i)}$ units of time, the activity must be executed at least once.

In the remainder of this paper, we will denoted by $\lceil Z[\overline{GSPS}] \rceil$ the smaller integer greater than or equal to the value optimal value of the LP-relaxation of $GSPS$ ("strong formulation").

The following proposition addresses the lower bound provided by the LP-relaxation of the "strong formulation".

*Proposition 4:* If the time horizon $H$ is greater or equal than the lowest common multiplier of the $\overline{F}_j$ $(j = 1, ..., J)$, then $\overline{x}$ defined by $\overline{x_{it}} = \dfrac{1}{\overline{F}_{j(i)}} \quad \forall i = 1, ..., n, \ t = 1, ..., H,$ is an optimal solution of the LP-relaxation of the "strong formulation" and, therefore, its optimal value is $\overline{R} = \sum_{j=1}^{J} \dfrac{n_j}{\overline{F}_j}$.

*Proof:* Let us show that $(\overline{R}, \overline{x})$ is feasible for the linear relaxation of the initial problem.

- $\sum_{i=1}^{n} \overline{x}_{it} = \sum_{j=1}^{J} \dfrac{n_j}{\overline{F}_j}$ and, hence, constraint (9) is satisfied by $(\overline{R}, \overline{x})$.
- $\sum_{l=t}^{t+\overline{F}_{j(i)}} \overline{x}_{il} = \dfrac{\overline{F}_{j(i)}}{\overline{F}_{j(i)}} = 1$ and constraint (12) is thus verified.
- $\sum_{l=t}^{t+\underline{F}_{j(i)}} \overline{x}_{il} = \dfrac{\underline{F}_{j(i)}}{\overline{F}_{j(i)}} \leq 1$ and constraint (11) is also verified.

Therefore, $(\overline{R}, \overline{x})$ is feasible for the linear relaxation. Let us now show that it is optimal. For this purpose, consider, for any (continuous) feasible solution $(R, x)$ of the linear relaxation the following quantity and any activity $i$ : $\sum_{t=1}^{F} x_{it}$ where $F$ is the lowest common multiplier of the $\overline{F}_j$. Then, from constraint 12, we have:

$$\sum_{t=1}^{F} x_{it} = x_{i1} + \ldots x_{i\overline{F}_{j(i)}} + x_{i\overline{F}_{j(i)}+1} \cdots + x_{iF} \geq \frac{F}{\overline{F}_{j(i)}} \tag{13}$$

since (12).

On the other hand, constraints (9) holds for any $t$ from 1 to $F$. Thus :

$$F \times R \geq \sum_{t=1}^{F} \sum_{i=1}^{n} x_{it} \tag{14}$$

and therefore (from(13):

$$F \times R \geq \sum_{j=1}^{J} F \frac{n_j}{\overline{F}_j} \tag{15}$$

that is

$$R \geq \sum_{j=1}^{J} \frac{n_j}{\overline{F}_j} = \overline{R} \tag{16}$$

It follows that the value provided by the objective function using any feasible solution is at least the value given by $\overline{x}$. As a consequence, $(\overline{R}, \overline{x})$ is optimal for the LP-relaxation of the "strong formulation" whenever the horizon is greater or equal to $F$. ■

*Corollary 1:* If the time horizon $H$ is greater or equal than the lowest common multiplier of the $\overline{F}_j$, then the two formulations are well named, that is the lower bound provided by the "weak formulation" is smaller or equal to the bound provided by the linear relaxation of the "strong formulation".

*Proof:* It is sufficient to establish that $(\overline{R}, \overline{x})$ is feasible for the "weak formulation", or in other words, that $\overline{x}$ satisfies constraint (10), using a simple calculation. ■

*Corollary 2:* If the time horizon $H$ is greater or equal than the lowest common multiplier of the $\overline{F}_j$ $(j = 1, ..., J)$, then the linear relaxation bound of the "strong formulation" is equal to the trivial lower bound.

*Proof:* Straightforward from propositions 2 and 4. ■

## VI. The proposed heuristics

We propose in this Section, three greedy algorithms so as to obtain good feasible solutions for $GSPS$. The main idea of the three heuristics is based on both the trivial lower bound and the possibility to schedule a task as late as it can be done. We successively give the main idea of the three greedy algorithms.

*Heuristics I and II.* First of all, Heuristics I and II only differ by a sorting criteria of the maximum periods. For both heuristics, the idea is to schedule first the most difficult types. The Heuristic I and II correspond to two different measures of what is a "difficult type". In Heuristic I, a type can be considered difficult if it is supposed to induce a large consumption of resources, which can be measured, from Proposition 2, by $\frac{n_j}{\overline{F_j}}$. Consequently, in Heuristic I, the types are sorted by decreasing order of the $\frac{n_j}{\overline{F_j}}$ whereas in Heuristic II, a type is "difficult" if we have only few possibilities to schedule a task after having chosen its first execution, that is if $\overline{F_j} - \underline{F_j}$ is "small". Thus, in Heuristic II, the types are sorted by increasing order of $\overline{F_j} - \underline{F_j}$. An outline of those two heuristics is given by Algorithm I and II.

**Algorithm 1** Heuristic I

```
Sort the types according to decreasing $\frac{n_j}{\overline{F_j}}$
for $j = 1$ to $J$ do
    Compute TLB for the first $j$ types
    if it is possible then
        Schedule successively the activities of type $j$ as late as possible within the open resources
    else
        open a new resource
    end if
end for
```

*Heuristic III.* We keep for this heuristic the sorting criteria relative to Heuristic II. Based on this we first schedule the tasks concerning the "easier" type and solve exactly the corresponding sub-problem on the whole horizon, using a branch-and-bound algorithm provided by Cplex10.0. We repeat this previous approach for all the types. The main steps of Heuristic III are reported in Algorithm 3.

**Algorithm 2** Heuristic II

```
Sort the types according to increasing $\overline{F_j} - \underline{F_j}$
for $j = 1$ to $J$ do
    Compute TLB for the first $j$ types
    if it is possible then
        Schedule successively the activities of type $j$ as late as possible within the open resources
    else
        open a new resource
    end if
end for
```

**Algorithm 3** Heuristic III

```
Sort the types according to increasing $\overline{F_j} - \underline{F_j}$
for $j = 1$ to $J$ do
    Successively schedule on the whole horizon each of the activities of the current type by minimizing R taking into account the activities already scheduled.
end for
```

## VII. Computational results

The objective of the computational study we conducted in this paper is to determine and/or assert:

- if the "strong formulation" allows a solver like ILOG-Cplex10.0 to solve to optimality large scale instances;
- if the lower bound ($\lceil Z[\overline{GSPS}]\rceil$) provided by the use of the "strong formulation" is of good quality and obtained quickly;
- if our (TLB) provides a good lower bound for $GSPS$ for both if $H$ is greater than the lower common multiplier of the maximum delays or not;
- the quality of the three upper bounds obtained utilizing the three algorithms we suggest, comparing those performance between each other and between the lower bounds;
- the approaches (I), (II) and (III) we proposed behave well and in a very fast CPU time.

Since no benchmark for $GSPS$ is available nowadays, we consider different two major types of randomly generated instances endowing each a particular structure:

1) ($H < lcm$): $H$ is lower than the lower common multiplier (lcm) of the maximum delays;
2) ($H \geq lcm$): $H$ is greater than or equal to the lower common multiplier (lcm) of the maximum delays. Actually, we set $H = lcm + 1$ (otherwise (lcm) takes very large values).

The rationale for utilizing instances (1) and (2) stems from the theoretical study of $GSPS$ which shows that the lower common multiplier (lcm) of the maximum period plays a key role in the evaluation of lower bound and in the resolution of $GSPS$.

In addition with regard to (1) and (2) problems, parameters $\overline{F_j}$, $\underline{F_j}$ and $n_j$ were respectively uniformly drawn at random in the range $\{1, ..., 20\}$, $\{1, ...\overline{F_j}\}$ and $\{1, ..., 20\}$ (cf. (1)) and such that $\{1, ..., 10\}$, $\{1, ...\overline{F_j}\}$ $\{1, ..., 10\}$ (cf. (2)). We reduce to $10$ the integer value possibility concerning instances (2) because of the largest value of the corresponding (lcm) which implies a very large value for $H$. Indeed, $H$ is a parameter which depends on the value of (lcm) which is computed after generation of the maximum delays. We also generated two sub-types of instances as follows:

(1.a) $(H < lcm)$ and all the parameters $\overline{F_j}$ and $\underline{F_j}$ are randomly generated in the range defined previously;
(1.b) $(H < lcm)$ and a half of the parameters $\overline{F_j}$ and $\underline{F_j}$ are equals and otherwise generated basically;
(2.a) $(H \geq lcm)$ and all the parameters $\overline{F_j}$ and $\underline{F_j}$ are randomly generated in the range defined previously;
(2.b) $(H \geq lcm)$ and a half of the parameters $\overline{F_j}$ and $\underline{F_j}$ are equals and otherwise generated basically.

The coefficient $J$ (i.e the number of types), is given. We chose to get $J = 5$ and $J = 10$. Consequently, the total number of the tasks equal to $\sum_{j=1}^{J} n_j$ is obtained by a simple calculation. On average concerning instances (1), (2) the number of total tasks is respectively equal to $55$, $30$ (cf. $J = 5$) and $106$, $120$ (cf. $J = 10$). We also play on the gap between the values of $\overline{F_j}$ and $\underline{F_j}$ so as to create more difficult instances. Indeed, smaller the gap is so more difficult is $GSPS$ to be solved to optimality in a fast CPU time (see instances (1.b et 2.b)). For example, the relative gap between the maximum and minimum delays is on average equal to $4$ for (1.a) and (2.a) where as it is on average equal to $2$ with regard to instances (1.b) and (2.b) when $\overline{F_j} \neq \underline{F_j}$ otherwise the gap is equal to $0$. In addition, $H$ is approximatively equal to $220$ for the simulations (1) where as its value is on average $400$ for problems (2). Finally, the size of problem $GSPS$ depends on the quantity of the tasks by type and on the size of the horizon time. Consequently the number of variables and contraints of $GSPS$ can be easy computed to give an idea of the difficulty of problems treated. For example, concerning instances (1.a) and (1.b) the problem size is of $12100$ and $23320$ variables whereas with regard to instances (2.a) and (2.b) the number of variables of $GSPS$ is equal to $12000$ and up to $48000$ variables. Thereby, we consider here very large scale problems which suggest that they would be difficult to be solved exactly using ILOG-Cplex10.0, especially for instances (2).

To assess the quality of the two lower bounds (LB) (namely, (TLB) and $\lceil Z[\overline{GSPS}]\rceil$) and the three upper bounds (UB) (cf. instances (1)) we used the optimum (when it can be found) to compute the relative gap (Gap = (UB - optimum)/(UB) or (optimum - LB)/(optimum)). Nevertheless, as the size of the problem $GSPS$ dramatically increases when $H \geq lcm$, the optimum is almost never reached by the use of the branch-and-bound provided by ILOG-Cplex10.0. Consequently, in the context of instances (2) we compared the lower bounds with the value of the upper bound we proposed i.e. Gap = (UB - LB)/(UB).

Our lower (TLB) and upper bounds were coded in C++ langage. The lower bound $\lceil Z[\overline{GSPS}]\rceil$ relative to the LP-relaxation of $GSPS$ as well as the optimum value of $GSPS$ were obtained using the commercial solver ILOG-Cplex10.0. Simulations were run on a bi-weon 3.4 GHz with 4 GB of main memory.

Table I displays the average deviation of each bound to the optimum over ten replications of each types of instances (1). It appears that our (TLB) as well as $\lceil Z[\overline{GSPS}]\rceil$ on average always reach the optimal value. It could be surprising because $H < lcm$ for these instances. Nevertheless, those two lower bounds behave very well even if when the gap between the maximum and minimum delays is small or equal to $0$ (cf. (1.b)). Concerning the feasible solutions, the lowest gap is obtained by Heuristic III which suggests that both sorting the type according to the lowest gap between the maximum and minimum periods and solving exactly each sub-problem corresponding to this selected type provide a feasible solution of good quality. However, Heuristic II (which utilizes the same previous sorting criteria) behaves less well than Heuristic I. Consequently, the only use of criteria relative to Heuristic II would not be sufficient to imply an upper bound closed to the optimum. Finally, for this type of problems (1), the “strong formulation” allows the branch-and-bound of ILOG-Cplex10.0 to solve to optimality most of the instances. Indeed, over $40$ replications, only two were not solved in a time limit equal to $10800$ seconds.

TABLE I
COMPARISON OF THE QUALITY OF THE UPPER AND LOWER BOUNDS WHEN OPTIMUM IS OBTAINED: INSTANCES (1)

| Inst. | # types | (TLB) | H.I | H.II | H.III | $\lceil Z[\overline{GSPS}]\rceil$ |
|---|---|---|---|---|---|---|
| | | Gap (%) | | | | |
| (1.a) | 5 | 0.0 | 9.49 | 22.86 | 5.05 | 0.0 |
| (1.b) | 5 | 0.0 | 10.2 | 25.1 | 7.3 | 0.0 |
| (1.a) | 10 | 0.0 | 12.68 | 28.69 | 5.85 | 0.0 |
| (1.b) | 10 | 0.0 | 13.2 | 29.62 | 6.02 | 0.0 |

Table II is concerned with the average deviation of each lower bound to the upper bounds over ten replications of each types of instances (2). Since, $H$ is greater than the lower common multiplier of the $\overline{F_j}$, its value is very large which implies that the problem size of $GSPS$ model also corresponds to very large instances. Indeed, $GSPS$ is an integer linear program which is well known to be difficult to solve in practice. Moreover, even if Heuristic III still provides the lowest gap, we note that Heuristic I behaves also well. In addition, concering the lower bounds, (TLB) and $\lceil Z[\overline{GSPS}]\rceil$ provide the same lower bound of quite good quality which asserts the theoretically result presented in Corollary 2.

Table III displays the CPU time in seconds required to compute the two lower bounds, the three upper boundsand the optimum (using ILOG-Cplex10.0) concerning instances (1) and (2). The advantage of using greedy heuristics strikingly appears: the branch-and-bound algorithm requires on average at most $291$ seconds to reach the optimum value. In any case, (TLB) and the two feasible solutions given by Heuristic I and II require less than one second to provide a value. The most time consuming bound is the LP-relaxation with a maximum

TABLE II
COMPARISON OF THE QUALITY OF THE UPPER AND LOWER BOUNDS WHEN OPTIMUM IS NOT REACHED: INSTANCES (2)

| Inst. | # types | (TLB) | | | $\lceil Z[\overline{GSPS}]\rceil$ | | |
|---|---|---|---|---|---|---|---|
| | | H.I | H.II | H.III | H.I | H.II | H.III |
| (2.a) | 5 | 8.93 | 10.17 | 6.16 | 8.93 | 10.17 | 6.16 |
| (2.b) | 5 | 9.07 | 11.18 | 7.02 | 9.07 | 11.18 | 7.02 |
| (2.a) | 10 | 21.65 | 40.75 | 3.53 | 21.65 | 40.75 | 3.53 |
| (2.b) | 10 | 22.01 | 41.03 | 4.08 | 22.01 | 41.03 | 4.08 |

of about 987 seconds to solve one of the harder instances (2.b). The time to compute the feasible solution provided by Heuristic III deviates at most of $36.67$ seconds for the largest problems (2.b). As a result, even if the quality of the upper bound obtained by Heuristic III is better than the two others, it takes a running time almost important which will not be advised to be used in a branch-and-bound algorithm. Finally, $\lceil Z[\overline{GSPS}]\rceil$ is very time consuming (up to $987$ seconds). Since in this particular instances, (TLB) and $\lceil Z[\overline{GSPS}]\rceil$ provides the same lower bound, it would be an advantage to utilize (TLB) instead of $\lceil Z[\overline{GSPS}]\rceil$ because of its faster running time.

TABLE III
COMPARISON OF THE CPU TIMES EXECUTION OF UPPER AND LOWER BOUNDS METHODS AND BRANCH-AND-BOUND ALGORITHMS: INSTANCES (1) AND (2)

| Inst. | # types | (TLB) | H.I | H.II | H.III | $\lceil Z[\overline{GSPS}]\rceil$ | Opt. |
|---|---|---|---|---|---|---|---|
| | | CPU times (s) | | | | | |
| (1.a) | 5 | 0.0 | 0.0 | 0.0 | 3.95 | 152.95 | 291.77 |
| (1.b) | 5 | 0.0 | 0.0 | 0.0 | 4.02 | 194.51 | 321.69 |
| (1.a) | 10 | 0.0 | 0.0 | 0.0 | 4.62 | 153.08 | 852.23 |
| (1.b) | 10 | 0.0 | 0.0 | 0.0 | 5.07 | 201.32 | 978.36 |
| (2.a) | 5 | 0.0 | 0.0 | 0.0 | 19.08 | 263.11 | - |
| (2.b) | 5 | 0.0 | 0.0 | 0.0 | 21.23 | 278.34 | - |
| (2.a) | 10 | 0.0 | 0.0 | 0.0 | 33.02 | 949.52 | - |
| (2.b) | 10 | 0.0 | 0.0 | 0.0 | 36.67 | 987.12 | - |

## VIII. CONCLUSION

In this paper we have studied $GSPS$ which is a generalization of the strictly periodic scheduling problem. After giving some properties, we have designed a simple technique to compute a lower bound (TLB) of good quality and obtained immediately. We also have proposed two mathematical formulations for $GSPS$. We have shown that "strong formulation" is more appropriated to solve $GSPS$. The lower bound derived from the resolution of the LP-relaxation of $GSPS$ "strong formulation" gives a value equal to (TLB) if $H \geq lcm$ and of good quality in any case. Finally, three greedy algorithms are suggested to get feasible solutions. Heuristic I and II provide upper bound of less quality than Heuristic III. Nevertheless, Heuristic III is very time consuming. A possible way to get further improvement of the computation time of the best feasible solution, would be to generate valid cuts so as to accelerate the resolution of each sub-problem.

## REFERENCES

[1] M. H. Ammar and J. W. Wong, *On the optimality of cyclic transmission in teletext systems*, IEEE Transactions on Communication, COM-35 1, 68–73, 1987.

[2] M. H. Ammar and J. W. Wong, *The design of teletext broadcast cycles*, Performance Evaluation 5 (4), 235–242, 1985.

[3] J. Anderson and P. Holman and A. Srinivasan *Fair scheduling of real time tasks on multiprocessors*, Handbook of scheduling : Algorithms, Models and Performance analysis, 31.1–31.21, J. Leung, ed., 2004.

[4] S. Anily and C. A. Glass and R. Hassin, *Scheduling of maintenance services to three machines*,Annals of Operations Research 85, 375–391, 1999.

[5] A. Bar-Noy and V. Dreizin and B. Patt-Shamir, *Efficient algorithm for periodic scheduling*, Computer Networks 45, 155–173, 2004.

[6] A. Bar-Noy and B. Randeep and J. Naor and B. Schieber, *Minimizing service and operation cost of periodic scheduling*, 9th ACM Symposium on Discrete Algorithms, 11–20, 1998.

[7] S. Baruah and J. Gehrke and C. G. Plaxton and I. Stoica and H. Abdel-Wahab and K. Jeffay, *Fair on-line scheduling of a dynamic set of tasks on a single resource*, Information Processing Letters 64 (1), 43–51, 1997.

[8] S. Baruah and N. Cohen and C. G. Plaxton and D. Varvel, *Proportionate progress: a notion in resource allocation*, Algorithmica 15 (6), 600–625, 1996.

[9] S. Baruah and J. Gehrke and C. G. Plaxton, *Fast scheduling of periodic traks on multiple resources*, 9th International Parallel Processing Symposium, 280–288, 1995.

[10] A. M. Campbell and J. R. Hardin, *Vehicle minimization for periodic deliveries*, European Journal of Operational Research 165, 668–684, 2005.

[11] M. Gaudioso and G. Paoletta, *A heuristic for the periodic vehicle routing problem*, Transportation Sciences 26, 86–92, 1992.

[12] M. Gaudioso and G. Paoletta and S. Sanna, *Management of periodic demands in distribution systems*, European Journal of Operational Research 20, 234–238, 1985.

[13] M. B. Jones and D. Rosu and M-C. Rosu, *Cpu reservations and time constraints: efficient, predictable scheduling of independant activities*, 16th ACM Symposium on Operating Systems Principles, 198–211, 1992.

[14] V. Kats and E. Levner, *Minimizing the number of vehicles in periodic scheduling: the non-euclidean case*, European Journal of Operational Research 107, 371–377, 1998.

[15] V. Kats and E. Levner, *Minimizing the number of robots to meet a given cyclic schedule*, Annals of Operations Research 69, 209–226, 1997.

[16] C. Kenyon and N. Schabanel and N. Young, *Polynomial-time approximation scheme for data broadcast*, 32nd Annual ACM Symposium on Theory of Computing, 659–666, 2000.

[17] C. Kenyon and N. Schabanel, *The data broadcast problem with non-uniform transmission times*, 10th Annual ACM Symposium on Discrete Algorithms, 547–556, 1999.

[18] S. Khanna and S. Zhou, *On indexed data broadcast*, 30th Annual ACM Symposium on Theory of Computing, 463–472, 1998.

[19] J. Korst, *Periodic multiprocessor scheduling*, Ph. D. thesis, Technical report University of Eindhoven, 1992.

[20] C. J. Liao and W. J. Chen, *Single-machine scheduling with periodic maintenance and nonresumable jobs*, Computers and Operations Research 30, 1335–1347, 2003.

[21] C. L. Liu and J. W. Layland, *Scheduling algorithms for multiprocessors in a hard-real-time environment*, Journal of the ACM 20 (1), 46–61, 1973.

[22] M. J. Negreiros and A. E. Xavier and A. F. Xavier and P. Michelon, *A framework of computational systems and optimization models for the prevention and combat of dengue*, International Transactions in Operations Research, to appear.

[23] K. S. Park and D. K. Yun, *Optimal scheduling of periodic activities*, Operations Research 33, 690–695, 1985.

[24] W. F. J. Verhaegh and P. E. R. Lippens and E. H. L. Aarts and J. L. van Meerbergen and A. van der Werf, *The complexity of multidimensional periodic scheduling*, Discrete Applied Mathematics 89, 213–242, 1998.

[25] W. Wei and C. L. Liu, *On periodic maintenance problem*, Operations Research Letters 2, 90–93, 1983.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Visualizing Multi-Dimensional Pareto-Optimal Fronts with a 3D Virtual Reality System

Elina Madetoja,
Henri Ruotsalainen,
Veli-Matti Mönkkönen
and Jari Hämäläinen
University of Kuopio
Department of Physics
P.O. Box 1627
FI-70211 Kuopio, Finland
Emails: {elina.madetoja,
henri.ruotsalainen, veli-matti.monkkonen,
jari.hamalainen}@uku.fi

Kalyanmoy Deb
Helsinki School of Economics
Department of Business Technology
P.O. Box 1210
FI-00101 Helsinki, Finland
Email: kalyanmoy.deb@hse.fi

***Abstract*—In multiobjective optimization, there are several targets that are in conflict, and thus they all cannot reach their optimum simultaneously. Hence, the solutions of the problem form a set of compromised trade-off solutions (a Pareto-optimal front or Pareto-optimal solutions) from which the best solution for the particular problem can be chosen. However, finding that best compromise solution is not an easy task for the human mind. Pareto-optimal fronts are often visualized for this purpose because in this way a comparison between solutions according to their location on the Pareto-optimal front becomes somewhat easier. Visualizing a Pareto-optimal front is straightforward when there are only two targets (or objective functions), but visualizing a front for more than two objective functions becomes a difficult task. In this paper, we introduce a new and innovative method of using three-dimensional virtual reality (VR) facilities to present multi-dimensional Pareto-optimal fronts. Rotation, zooming and other navigation possibilities of VR facilities make easy to compare different trade-off solutions, and fewer solutions need to be explored in order to understand the interrelationships among conflicting objective functions. In addition, it can be used to highlight and characterize interesting features of specific Pareto-optimal solutions, such as whether a particular solution is close to a constraint boundary or whether a solution lies on a relatively steep trade-off region. Based on these additional visual aids for analyzing trade-off solutions, a preferred compromise solution may be easier to choose than by other means.**

## I. Introduction

IN MANY real-world problems, decision making with multiple conflicting objectives in every day operation can be demanding. Moreover, an unfavorable decision can be financially expensive or even hazardous in some situations. There exist different ways to support a decision making process [1], and some of them are based on multiobjective optimization. Multiobjective optimization methods are capable of handling multiple conflicting objectives at the same time. Solutions of the multiobjective optimization problem form a Pareto-optimal front, i.e. a set of compromised trade-off solutions. However, even when different Pareto-optimal solutions are found, choosing a particular optimal compromise solution is not a trivial task. Furthermore, the objectives in a multiobjective optimization task do not need to be commensurable. In such a case, the multiobjective decision making task gets more difficult, especially when the number of objectives is larger than two. This is why there is a need for developing new methodologies for supporting the decision making process.

A Pareto-optimal front, from where an individual final solution can be chosen (e.g. by a decision maker [2]) is often studied with different visualization tools. In this way, the decision maker can extract useful information from the results and thus, a comparison between solutions gets easier. A Pareto-optimal front is quite simple to visualize when there are only two objective functions. However, visualizing with more than two objectives has so far been problematic, and few attempts have been made to visualize a higher dimensional Pareto-optimal front [3], [4], [5]. Virtual reality (VR) is a visualization environment that offers facilities to present high-dimensional spaces and it has also been applied for Pareto-optimal fronts, see [6], [7], [8]. In this paper, we suggest the use of the VR environment not only to visualize a higher dimensional Pareto-optimal front, but also to analyze and understand the nature and relative location of solutions in order to help choosing the best solution for the particular problem.

Basically, the VR is a computer created environment which can be used for visualizing three-dimensional (3D) objects (see, e.g. [9], [10]). Hence it makes possible to visualize and compare solutions which are on a 3D Pareto-optimal front. Thus, a visualized 3D Pareto-optimal front can be examined in many ways: it can be zoomed and rotated, and it also allows the decision maker to dive into the front to get a feel of the nature of the solutions. Moreover, VR enables the user to interact with the visualized Pareto-optimal solutions. This makes easier to compare neighboring solutions and allows the decision maker to learn about the problem and

978-83-60810-14-9/08/$25.00 © 2008 IEEE

the interrelationships among objectives. Based on these VR facilities the decision maker can identify a particular solution which is an adequate compromise.

In this paper, there are two case studies utilizing 3D VR facilities presented. First, we use an evolutionary computation -based multiobjective optimization scheme for generating a large number of Pareto-optimal solutions. In addition, an interactive visualization scheme in the VR is used to decipher some interesting features of the solutions obtained by using existing methodologies such as the concept of *innovization* [11]. Second, there is an industrial example having four conflicting objectives presented. Although we show a few capabilities of a VR system here for decision making purpose, certainly many other innovative methodologies are possible, and this paper should encourage execution of further studies in the coming years.

## II. Multiobjective Optimization and Virtual Reality

A multiobjective optimization problem is often defined as follows:

$$\begin{aligned}&\text{minimize}\,\{f_1(\boldsymbol{x}),\ldots,f_k(\boldsymbol{x})\}\\&\text{subject to }\boldsymbol{x}\in S,\end{aligned}\tag{1}$$

where $\boldsymbol{x}$ is a vector of decision variables from the feasible set $S \subset \mathbb{R}^n$ defined by linear, nonlinear and box constraints. An objective vector can be denoted by $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$. Here we minimize, but if an objective function $f_i$ is to be maximized, it is equivalent to consider minimization of $-f_i$.

Optimality in multiobjective optimization is understood in the sense of Pareto-optimality or non-dominated solutions [12]. The Pareto-optimality is defined as follows: *a decision vector $\boldsymbol{x}' \in S$ is Pareto-optimal if there does not exist another decision vector $\boldsymbol{x} \in S$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}')$ for all $i = 1, \ldots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}')$ for at least one index $j$.* These Pareto-optimal solutions form a Pareto-optimal set or a Pareto-optimal front. There are two concepts often used in multiobjective optimization: an ideal objective vector $z^* \in \mathbb{R}^k$ and a nadir objective vector $z^{nad} \in \mathbb{R}^k$ that give lower and upper bounds, respectively, for the objective functions in the Pareto-optimal front (see [2] for details). All the Pareto-optimal solutions are equally good compromises from a mathematical point of view, and there exists no trivial mathematical tool to find the best solution in the Pareto-optimal front. Typically a decision maker, who is an expert in the field from where the problem has arisen, is needed in order to find the best or the most satisfying solution. The decision maker can participate in the process of finding the solution in the different ways and also the different phases of solving process by determining which of the Pareto-optimal solutions is the most satisfying to be the final solution. However, decision making is sometimes tricky, because comparing the numerical values of the solutions is difficult. Thus, some additional information and aids are needed to support decision making process.

### A. Virtual Reality Environment

Virtual reality is a medium which makes it possible to visualize and experience objects from an animated world having visual, sound and haptic experiences realized through immersion, interaction, and collaboration of the VR elements. According to [10] four key elements create the VR environment: a virtual world, immersion, sensory feedback, and interactivity. *The virtual world* is a content of given medium including a collection of objects, and their relationships and rules in the space. *Immersion* into an alternative reality means possibility to perceive something besides the world in which one is living currently. Immersion is sometimes divided into physical and mental immersions, but often they both exist in a VR system. *The sensory feedback* is the third key element. It is based on user's physical position, and the aim is that the objects and the whole space alter depending on the user's position. The last element is *interactivity* which means real-time response to the user's actions. There are many applications that can utilize virtual reality technology: visualizing scientific results, interior design in architecture, and prototype testing in industry [13], [14], for example.

In the VR laboratory objectives can be examined in their real size or small objectives can be enlarged, which makes VR usable in several applications. VR can be also build in PC environment with feasible equipment, software, active stereo glasses etc. Then navigation is not comprehensive as in laboratory, but the user is still able to interact with VR similarly to laboratory environment.

### B. 3D Virtual Reality Utilized in Multiobjective Optimization

The VR system can be used in visualizing Pareto-optimal solutions, and thereby supporting the decision making in a multiobjective optimization process. The flexibility associated with a 3D VR system makes it an alternative way of visualizing the Pareto-optimal front and suing visual information in the decision making process. In addition, analyzing the solutions in order to understand the interactions of objective functions and decision variables comes easier. In Fig. 1 the users utilize a VR environment in order to examine an approximated Pareto-optimal front. The front is controlled (zoomed, rotated and scaled) by the user using a wand (also called a 3D-mouse). In the VR, the users can study the relationships between objective functions and then get ideas what kind of compromises between the multiple objectives can be made. This process will then aid in selecting the final compromised solution. When complexity of the data increases, valuable information of the Pareto-fronts and problem's behaviour can be extracted from graphical presentation efficiently. One should note that the real immersion and 3D objects can be experienced only in a virtual reality laboratory, not in the figures presented in this paper.

Integration of multiobjective optimization and VR requires a computerized algorithm for calculating Pareto-optimal solutions and a hardware system for a VR environment coupled with a software for visualization in stereo [10]. The VR environment used in this research has been built at the University of Kuopio in Finland, and it is based on OpenDX visualization

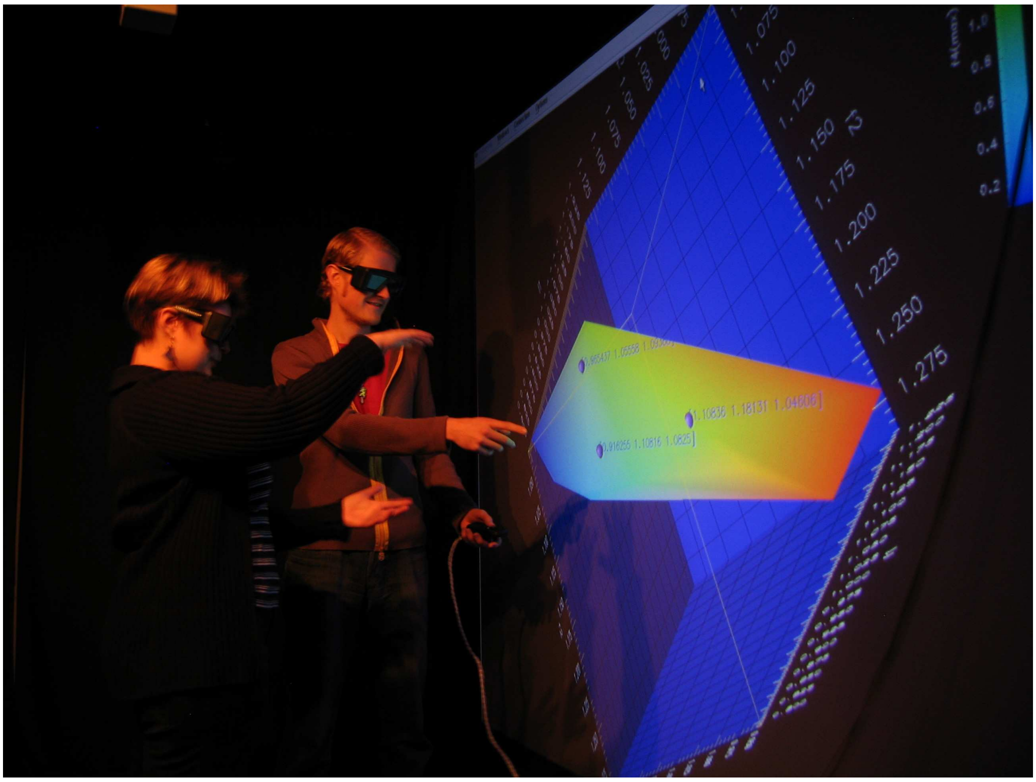


Fig. 1. Analyzing a Pareto-optimal front with the 3D VR system. A 3D Pareto-optimal front visualized in the VR system is presented in a two-dimensional figure here, because the real immersion on a front can be experienced only in a laboratory.

software. Graphics Computer SGI Prism with 8 CPUs (a 64-bit, 1.5 GHz, Intel Itanium 2, 24 GB memory, 48 Gflops) is used with SuSe Linux Enterprise Server 9.3 as an operating system. 3D-effects are generated through wireless set of liquid crystal shutter eye wear (active stereo glasses). Stereo glasses shut alternately left and right eye view with frequency about 45 pictures per eye per second. Visualized 3D-objects (in this paper Pareto-optimal fronts) are controlled through the wand. Polhemus equipment is used to follow the wand's movements to control the objects in the VR environment.

## III. Visualization Examples

In this section, we present two examples illustrating the new visualizing aspects which a VR facility can offer. In the first example, a standard three-objective test problem having a disconnected set of non-linear Pareto-optimal fronts was solved by evolutionary multiobjective optimization (EMO). EMO procedures are generic population-based meta-heuristic optimization algorithms [12]. They use natural evolutionary principles, such as reproduction, mutation and recombination, iteratively to attempt to find a set of Pareto-optimal solutions. EMO methodologies are capable of finding a large set of trade-off solutions as presented in the first example.

In the second example, we consider a real-world paper-making optimization problem with four objective functions in which the advantages of decision making aspects with the VR system are presented. This example was solved by a classical multiple criteria decision making method. The solution process contained two steps, and in the first step a genetic algorithm with a scalarizing function [2] was used. Then neighboring solutions were connected with a hyper-plane in visualization, and any point in the hyper-plane could be chosen as a reference point. Thus, in the second step the corresponding Pareto-optimal solution could be obtained by solving an achievement scalarizing function [2], [5], [15].

### *A. A Test Problem with Highlighted Solutions*

First, we considered a multiobjective optimization test problem (DTLZ6) [16]. In the general form of this problem, there are $k$ objective functions with a complete decision variable vector partitioned in $k$ non-overlapping groups $\boldsymbol{x} \equiv (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)^T$. We solved a three-objective version of the problem that is written as follows [16]:

$$\begin{array}{ll} \text{minimize} & \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})\} \\ \text{subject to} & 0 \leq x_i \leq 1 \text{ for } i = 1, \ldots, 22, \end{array} \tag{2}$$

where the objective functions were defined as $f_1(\boldsymbol{x}_1) = x_1$, $f_2(\boldsymbol{x}_2) = x_2$ and $f_3(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_3))h(f_1, f_2, g)$. The functionals were $g(\boldsymbol{x}_3) = 1 + \frac{9}{|\boldsymbol{x}_3|}\sum_{x_i \in \boldsymbol{x}_3} x_i$ and $h(f_1, f_2, g) = 3 - \sum_{i=1}^{2}\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right]$. The functional $g(\boldsymbol{x}_3)$ required $|\boldsymbol{x}_3| = 20$ variables and $n$ was the total number of variables, here $n = 22$. In this test problem, there were $2^2 = 4$ disconnected Pareto-optimal regions.

The NSGA-II procedure [12] was used as an EMO algorithm in this study and it was run with 1,000 population

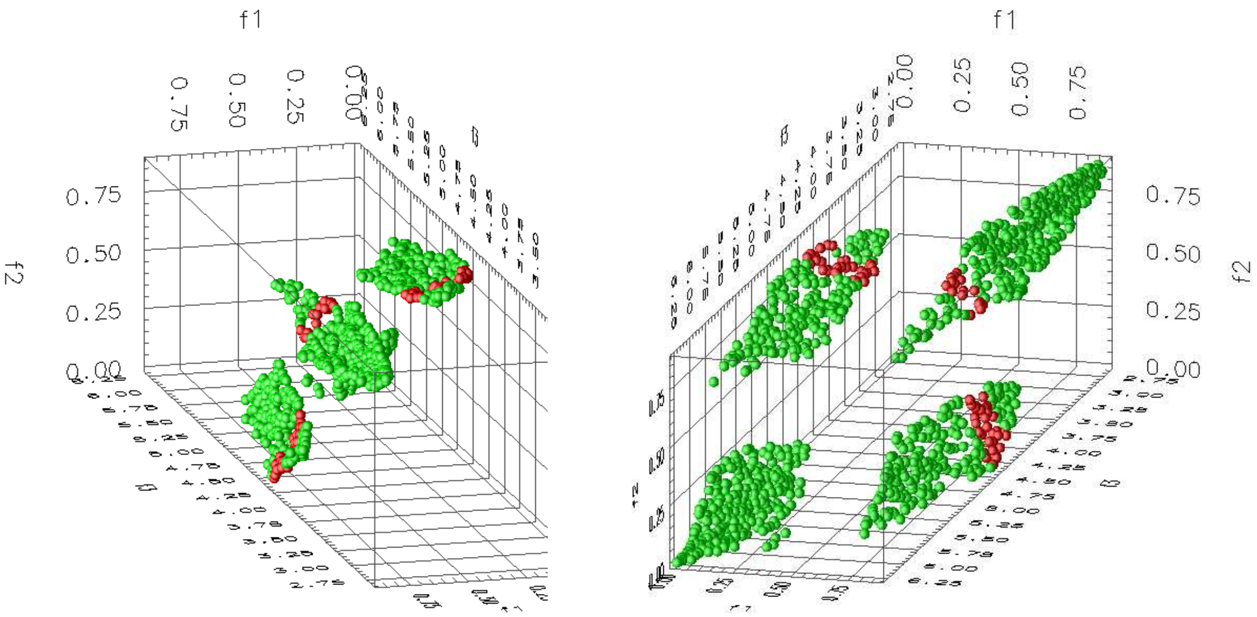


Fig. 2. Disconnected set of nonlinear Pareto-optimal regions (forming the Pareto-optimal front) in evolutionary computation example (DTLZ6) in a VR environment. Solutions having $4.2 \leq f_3 \leq 4.4$ are colored red.

members. The final solutions were visualized with the proposed VR system and they are shown also in Fig. 2. With the 3D visualization capabilities, the nonlinear feature of the disconnected Pareto-optimal regions was much easier to see compared to the earlier studies [16]. In this example, the decision maker was first interested in seeing all the solutions in which $4.2 \leq f_3 \leq 4.4$ as presented in Fig. 2. As one can see in the figure, this constraint made possible quite different compromises between the other two objectives: highlighted solution were located into three separate Pareto-optimal regions. Thus, understanding the trade-offs between different targets was more clear. Moreover, because of zooming, rotating, and immersion possibilities, the Pareto-optimal front was easy to comprehend. Also, the trade-offs between targets were easy to understand, and hopping from one Pareto-optimal region to another got simplified.

In the concept of *innovization* [11], the task of evolutionary multiobjective optimization is followed by a search of hidden interactions among decision variables and objective functions within obtained solutions. This concept has revealed interesting and important insights about design and optimization problems. Here, we argue that the proposed VR based visualization tool can be used as an aid to assist in the *innovization* task. Combining these two concepts allows the decision maker to test the validity of different interrelationships among the decision variables and objective functions. For example, the existence of a given relationship, such as $\Phi(\boldsymbol{f}, \boldsymbol{x}) = 0$, can be tested by marking all solutions (among the obtained EMO solutions) which restrict the absolute value of $\Phi$ within a threshold, say $\epsilon = 10^{-6}$, in red. The location and trace of these solutions on the Pareto-optimal front will provide a plethora of information to the decision maker about the importance of the above relationship before choosing a particular solution. To illustrate, we return to the DTLZ6 test problem and investigate the existence of Pareto-optimal solutions satisfying the following relationships:

$$\begin{aligned}
\Phi_1(\boldsymbol{f}, \boldsymbol{x}) &: x_1 = 0 \quad \text{(Red)},\\
\Phi_2(\boldsymbol{f}, \boldsymbol{x}) &: x_2 = 0 \quad \text{(Blue)},\\
\Phi_3(\boldsymbol{f}, \boldsymbol{x}) &: x_1 = 1 \quad \text{(Brown)},\\
\Phi_4(\boldsymbol{f}, \boldsymbol{x}) &: x_2 = 1 \quad \text{(Purple)}.
\end{aligned}$$

The above conditions check if any Pareto-optimal solution made box constraints on variables $x_1$ and $x_2$ active. In Fig. 3, there are marked all such solutions with $\epsilon = 10^{-6}$. It is interesting to note that there were no solutions on the Pareto-optimal front close to the upper bound of these two variables and there were a number of solutions which were close to their lower bounds. Only a few solutions made $x_1$ close to zero, but there exist a number of solutions which made $x_2$ close to zero. Furthermore, all these solutions seemed to lie on only one of the four Pareto-optimal regions. It could be useful to identify solutions close to constraint boundaries and a further investigation and relaxation of active constraints could lead to better solutions. Such information was not only interesting but could be useful if problem-specific relationships were tested.

As seen from this example, the VR environment can be used as a 3D visualization tool for Pareto-optimal solutions obtained with an EMO procedure. These solutions can be studied with a VR tool not only to make a better visualization of the front, but also to gather more useful information and properties of Pareto-optimal solutions. Next, we present a more complex real-world industrial decision making problem.

### B. Industrial Example: Papermaking Optimization

In papermaking, the aim is to produce paper as much as possible with as low costs as possible [17], [18], [19]. In addition, there are several quality properties which should

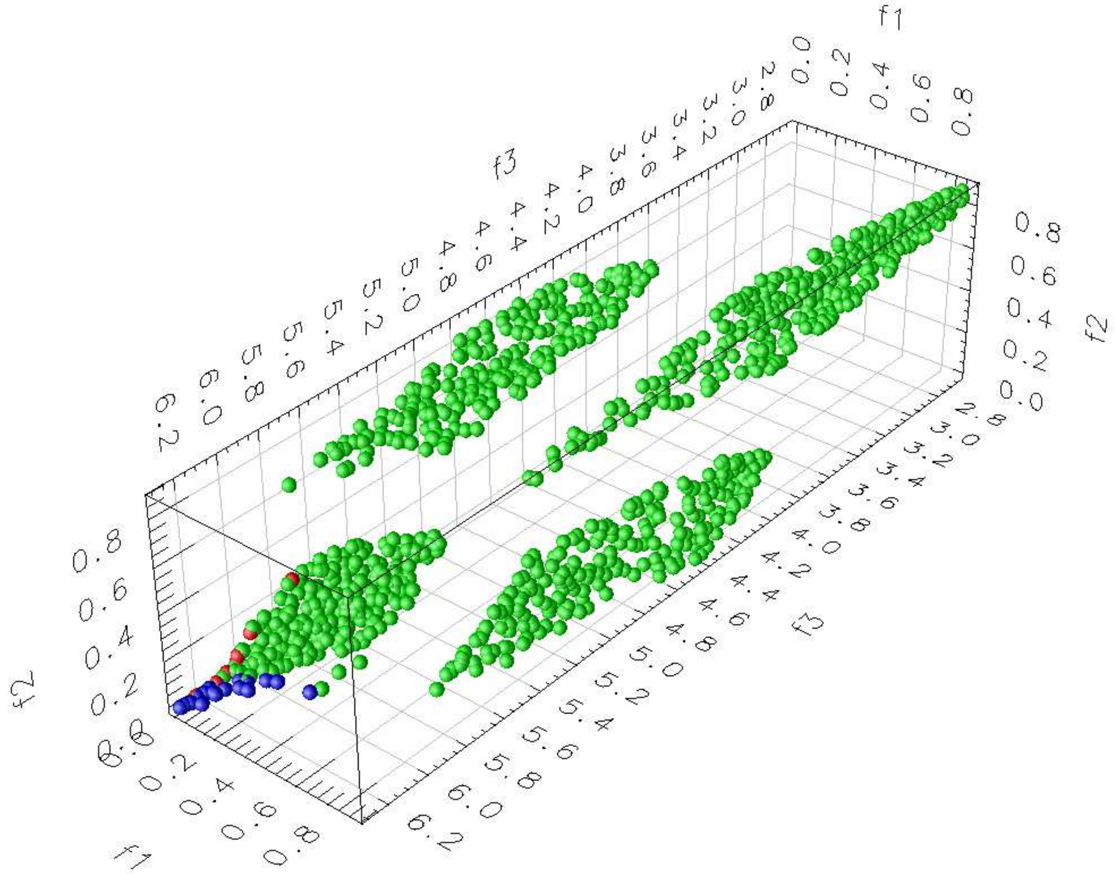


Fig. 3. Pareto-optimal solutions close to the constraint boundaries are highlighted with red and blue colors using the VR environment for the DTLZ6 test problem.

obtain acceptable values at the same time. These targets are often conflicting, and thus the optimization problems become naturally multiobjective. In this example a papermaking optimization problem is studied. Because of the long computational time, the decision maker wanted to compute as few solutions as possible. The Pareto-optimal solutions computed were visualized as 3D points in the VR, and an approximation of a Pareto-optimal front was formed with these few solutions.

In this example, there were four papermaking objectives and eight decision variables. The problem was formed as a model-based optimization problem [18], where the objective function values could be evaluated based on the solution of equations describing the system, i.e. a simulation model of a paper machine. Thus, the optimization problem was written as follows:

$$\begin{aligned} &\text{optimize } \{f_1(\boldsymbol{x},\boldsymbol{q}_1,\ldots,\boldsymbol{q}_{27}),\ldots,f_4(\boldsymbol{x},\boldsymbol{q}_1,\ldots,\boldsymbol{q}_{27})\} \\ &\text{subject to } \begin{cases} A_1(\boldsymbol{x},\boldsymbol{q}_1)=0 \\ A_2(\boldsymbol{x},\boldsymbol{q}_1,\boldsymbol{q}_2)=0 \\ \vdots \\ A_{27}(\boldsymbol{x},\boldsymbol{q}_1,\ldots,\boldsymbol{q}_{27})=0 \end{cases} \\ &\qquad\qquad \boldsymbol{x}\in S, \end{aligned} \tag{3}$$

where $f_1$ presented paper tensile strength ratio and it was given the desired value 3.4. The objective functions $f_2$ and $f_3$ were paper formation and basis weight, which were given the desired values 0.36 $g/m^2$ and 50.5 $g/m^2$, respectively. The fourth objective function $f_4$ was evaporated water which was to be maximized. A vector $\boldsymbol{x} \in S$ contained all the decision variables that were typical controls of paper machine and the feasible set $S$ was formed of their box constraints. Mappings $A_i$ for all $i = 1, \ldots, 27$ denoted unit-process models constituting a simulation model for the entire papermaking process, and $\boldsymbol{q}_i,\ i = 1, \ldots, 27$ were the simulation model outputs [18].

The optimization process contained two separate steps. *In the first step*, a set of the trade-off solutions were calculated with a genetic algorithm with scalarization by achievement scalarizing function. Then, an approximation of the Pareto-optimal front was generated in the VR environment using these solutions. The left plot in Fig. 4 shows the solutions and the approximated front obtained after the first step. The values of the objective functions $f_1, f_2$ and $f_3$ are presented on the axes and $f_4$ is presented by colour in the figure. Here, the proposed VR environment was found to be quite effective tool to explore the multidimensional Pareto-optimal solutions and the approximated front between them. The decision maker observed that there was a conflict between the first two objective functions, i.e. a good tensile strength ratio caused a large formation value which was not desired and vice versa. Thus, there exists a trade-off. Another observation was that a

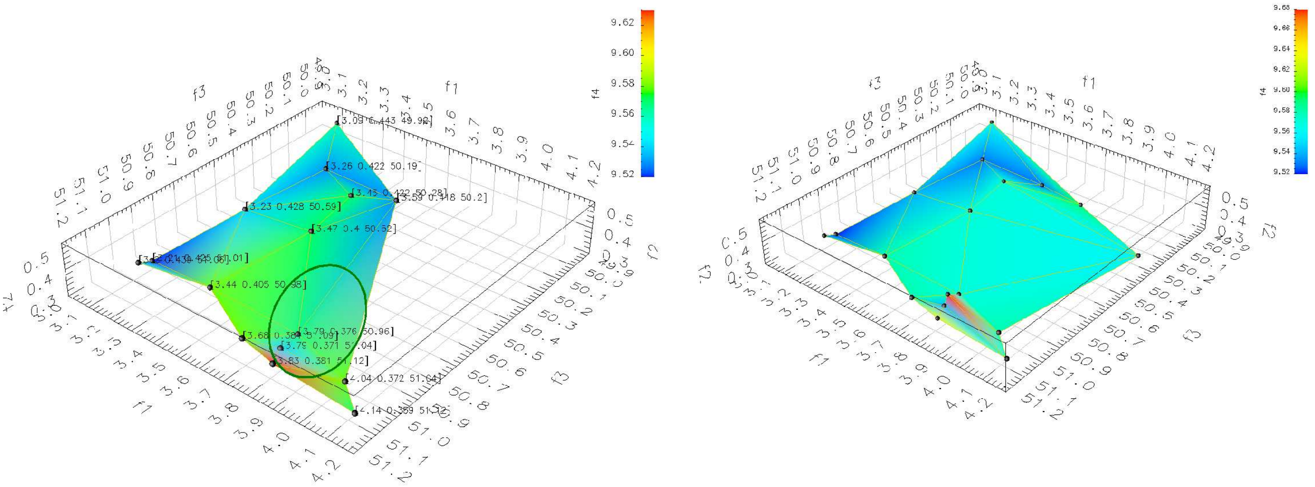


Fig. 4. On the left, solutions after the first step, and on the right, all solutions after the first and the second steps. The objective functions from $f_1$ to $f_3$ are presented as a 3D surface and $f_4$ is presented as a variation in color.

large value of the fourth objective function came with a large value of the third objective function, thereby producing also a conflict between these two objective functions: the desired value of $f_3$ could not be achieved at the same time with a good value of $f_4$. However, there were good compromise solutions between the objective functions on middle and front part of the approximated set. Based on these observations, the region highlighted by an ellipse (shown in the figure) was chosen and the optimization process was re-directed towards this region in the second step. This preference information was obtained with the help of visualization through the VR tool, where the decision maker could examine the existing solutions in many ways by rotating and zooming the Pareto-optimal set.

*In the second step*, three new solutions were calculated with help of the reference point method and the gradient-based optimizer. The decision maker's preferences, the circled region in Fig. 4, was utilized in defining the reference points. Unfortunately, only one of the three new solutions generated was located into the preferred region and other two were located in such a part of the solution space, where there were no solutions after the first step. The plot on the right side in Fig. 4 shows all the solutions, that is the solutions produced in the first step complemented by three solutions produced in the second step. Two of the new solutions were interesting from the papermaking point of view: one located inside the preferred region (circled in Fig. 4) and another one located on the right side having values ($\boldsymbol{f} = (3.78, 0.39, 50.19, 9.59)^T$), which presented also a good compromise between the objective functions. However, the first-mentioned solution (inside the circled part) had objective function values: $\boldsymbol{f} = (3.79, 0.41, 51.02, 9.68)^T$ and it was the most satisfying compromise solution to be the final one according to the decision maker's knowledge.

The ability to visualize trade-off information among objective functions through the 3D VR system makes it possible to focus on interesting part of the solution space. This will certainly enhance the decision making ability in computationally demanding real-world optimization problems and reduce the number of uninteresting solutions needed be calculated. In addition, better visualization technique allows one to get more information about the relationships between the solutions and objective functions than a simple plot of the numerical data. We believe that the VR tool will help the decision maker to understand and analyze the Pareto-optimal front, and thus make it easier to choose a single preferred solution.

## IV. Discussions and Conclusions

In multiobjective decision making, Pareto-optimal fronts are often visualized because in this way a comparison between solutions becomes easier. A Pareto-optimal front is easy to visualize when there are only two objective functions, but visualizing more than two objective functions is problematic. In this paper, we have integrated multiobjective optimization with the 3D VR tool to study the Pareto-optimal solutions and approximated Pareto-optimal fronts to help to make a better decision when choosing the final solution. The 3D VR tool makes easier to compare solutions, navigate from one solution to the other by zooming and rotating the front. Thus, it allows a better comprehension of solutions with desired properties through highlighting. In addition, using sophisticated visualization tools means that fewer solutions need to be computed in order to learn and understand the interrelationships of the conflicting objectives. This is important especially if a problem is computationally expensive (e.g. in real-world industrial cases).

In this paper, different kinds of 3D visualizations with the VR environment have been discussed and demonstrated. First, a large number of solutions forming a dense set of Pareto-optimal solutions obtained by EMO was visualized. In this set, some interesting features of the solutions were highlighted and studied. In the second problem, a few Pareto-optimal solutions were calculated with different optimization techniques and they were visualized using the proposed VR environment. The information gathered from this exercise helped to find an interesting Pareto-optimal region for the decision maker to concentrate. Such a technique will be valuable for handling large number of objective functions.

This paper, so far, has shown a number of advantages of using a VR environment in making a better realization of the Pareto-optimal front in a multiobjective optimization task. In addition, we have emphasized capabilities of VR that helps decision making in real-world applications, which we see as one of the potential application of the VR systems. These initial results are promising and open up a number of challenging research issues, such as handling a large number of objective functions, simultaneous visualization of objective and solution spaces, faster an optimization software and a VR hardware interactions, etc. The purpose of this paper is to discuss the power and usefulness of the VR environment in multiobjective optimization, and to bring out the technique as a new and promising mean of visualizing and understanding complex interactions among objectives and solutions.

## REFERENCES

[1] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*. North-Holland: Elsevier Science Publishing, 1983.

[2] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publisher, 1999.

[3] A. V. Lotov, "Approximation and visualization of Pareto frontier in the framework of classical approach to multi-objective optimization," in *Practical Approaches to Multi-Objective Optimization*, ser. Dagstuhl Seminar Proceedings, J. Branke, K. Deb, K. Miettinen, and R. E. Steuer, Eds., no. 04461, Dagstuhl, Germany, 2005.

[4] A. V. Lotov, L. V. Bourmistrova, R. V. Efremov, V. A. Bushenkov, A. L. Buber, and N. A. Brainin, "Experience of model integration and Pareto frontier visualization in the search for preferable water quality strategies," *Environmental modelling and software*, vol. 20, no. 2, pp. 243–260, 2005.

[5] A. V. Lotov, V. Bushenkov, and G. Kamenev, *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*, ser. Applied Optimization. Springer, 2004, vol. 89.

[6] E. Madetoja, H. Ruotsalainen, and V.-M. Mönkkönen, "New visualization aspects related to intelligent solution procedure in papermaking optimization," in *EngOpt 2008—International Conference on Engineering Optimization*, Rio de Janeiro, Brazil, 2008.

[7] J. Valdés and A. Barton, "Visualizing high dimensional objective spaces for multi-objective optimization: A virtual reality approach," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore, 2007.

[8] J. Valdés, A. Barton, and R. Orchard, "Virtual reality high dimensional objective spaces for multi-objective optimization: An improved representation," in *Proceedings of IEEE World Congress on Evolutionary Computation*, Singapore, 2007.

[9] J. Eddy and K. E. Lewis, "Visualization of multidimensional design and optimization data using cloud visualization," in *Proceedings of DETC'02, ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, Canada, 2002.

[10] W. Sherman and A. Craig, *Understanding Virtual Reality: Interface, Application, and Design*. San Francisco: Elsevier Science, 2003.

[11] K. Deb and A. Srinivasan, "Innovization: Innovating design principles through optimization." in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, New York, 2006, pp. 1629–1636.

[12] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, 2nd ed. Chichester: John Wiley & Sons, 2001.

[13] C. Antonya and D. Talaba, "Design evaluation and modification of mechanical systems in virtual environments," *Virtual Reality*, vol. 11, no. 4, pp. 275–285, 2007.

[14] S. Jayaram, U. Jayaram, Y. Kim, C. DeChenne, K. Lyons, C. Palmer, and T. Mitsui, "Industry case studies in the use of immersive virtual assembly," *Virtual Reality*, vol. 11, no. 4, pp. 217–228, 2007.

[15] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Applications*, F. G. and T. Gal, Eds. Berlin: Springer-Verlag, 1980, pp. 468–486.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, R. Jain, and R. Goldberg, Eds. Springer, 2005, ch. 6, pp. 105–145.

[17] J. Hämäläinen, T. Hämäläinen, E. Madetoja, and H. Ruotsalainen, "CFD-based optimization for a complete industrial process: Papermaking," in *Optimization and Computational Fluid Dynamics*, D. Thévenin and G. Janiga, Eds. Springer, 2008.

[18] E. Madetoja, "Novel process line approach for model-based optimization in papermaking—sensitivity and uncertainty analysis," Ph.D. dissertation, University of Kuopio, 2007.

[19] E. Madetoja, E.-K. Rouhiainen, and P. Tarvainen, "A decision support system for paper making based on simulation and optimization," *Engineering with Computers*, vol. 35, no. 5, pp. 461–472, 2008.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Communication Network Design Using Particle Swarm Optimization

C. Papagianni
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneioy 9 Str. 15773 Zografou, Athens, Greece

Email: chrisap@telecom.ntua.gr

K. Papadopoulos, C. Pappas,
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneioy 9 Str. 15773 Zografou, Athens, Greece

Email: kpapadop@esd.ece.ntua.gr, chrispap@telecom.ntua.gr

N. D. Tselikas,
D. T. Kaklamani, I. S Venieris
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneioy 9 Str. 15773 Zografou, Athens, Greece

Email: ntsel@telecom.ntua.gr, dkaklam@cc.ece.ntua.gr, venieris@cs.ntua.gr

*Abstract*—**Particle Swarm Optimization is applied on an instance of single and multi criteria network design problem. The primary goal of this study is to present the efficiency of a simple hybrid particle swarm optimization algorithm on the design of a network infrastructure including decisions concerning the locations and sizes of links. A complementary goal is to also address Quality of Service issues in the design process. Optimization objectives in this case are the network layout cost and the average packet delay in the network. Therefore a multi-objective instance of the hybrid PSO algorithm is applied. The particular hybrid PSO includes mutation to avoid premature convergence. For the same reason repulsion/attraction mechanisms are also applied on the single objective case. Mutation is passed on to the multi-objective instance of the algorithm. Obtained results are compared with corresponding evolutionary approaches.**

## I. Introduction

The increasing complexity of the network design problems calls for advanced optimization techniques. Network design problems where even a single cost function is optimized are often NP-hard [1]. In addition communication network design problems are not time critical. Therefore approaches have been designed to address these problems based in meta-heuristics such as simulated annealing, taboo search, evolutionary computing, nature inspired algorithms or both [2][3]. Concerning evolutionary computing in telecommunication network design, a comprehensive study is presented in [4] up to 2005 containing relevant research study references, where network design problems are classified in node location problems, topology design, tree design, routing, restoration, network dimensioning, admission control and frequency assignment/wavelength allocation. The optimization techniques employed are mainly variations of Genetic Algorithms. Additional work on telecommunication network optimization has followed in the last three years.

Real world network design problems normally involve the simultaneous optimization of multiple and usually partially contradicting objectives. Therefore more often than not, there is not a single optimal solution, given the diversity of the set of objectives, but a set of congruent solutions, known as Pareto-optimal. The topological design of communication networks is usually a multi-objective problem involving simultaneous optimization of the cost concerning network deployment as well as various performance criteria (e.g. average delay, throughput) subject to additional constraints (e.g. reliability, bandwidth). These problem specific objectives are often opposing; for example a way to reduce average delay in the network is over provisioning; that is to increase available link capacities which will consequently result in the increase of the total network deployment cost.

In this paper we will use Particle Swarm Optimization algorithm for the Topological Network Design problem, including capacity allocation, considering shortest path routing. Therefore the target is to design a near optimal network infrastructure, including decisions concerning the locations and sizes of links. For that purpose, a hybrid version of the PSO algorithm will be applied to the real network problem introduced by Rothlauf [5] and its efficiency will be evaluated against GAs. In addition a bi –criteria communication network topology problem is considered to address Quality of Service issues in the design process. For the corresponding delay function, a Poisson traffic model is utilized [1][3][7]. This real world application is addressed using multi-objective PSO. The Pareto front obtained by the MOPSO application is compared to the results obtained by a multi-objective GA (NSGA-II [6]). Relevant work on the subject has been presented in [1][7] among others using EAs. An alternative approach is also proposed in [8] where the relevant Delay Constrained Least Cost Path problem is addressed, utilizing the principle of Lagrangian relaxation based aggregated cost, where a PSO and noising metaheuristic are used for minimizing the modified cost function.

Of crucial importance to the success of the optimization procedure is the choice of candidate solutions representation. Especially for evolutionary algorithms a variety of encodings have been proposed as characteristic vectors, predecessors, Prufer numbers, link and node biasing, edge sets etc. In [8] [9] a tree based encoding/decoding scheme, based on heuristics has been devised for representing the paths as particles. In the presented work a tree is encoded with the network random keys (NetKeys) scheme introduced in [10].

978-83-60810-14-9/08/$25.00 © 2008 IEEE

In [11] Random keys were adapted to represent spanning trees. In this coding, a tree is represented by a string of real-valued weights, one for each edge of the complete graph. Therefore the size of the encoded string for a graph representing *N* nodes is *N∗(N-1)/2*. In order to decode the string, the edges are sorted by their weights and Kruskal's minimum spanning tree algorithm considers the edges in sorted order. The mapping is one to one since any string of weights is a valid tree. Using NetKeys, that represents a tree with normalized real values, allows the optimization algorithms to avoid making binary and hard decisions on whether to establish a link or not.

In Section II a short overview of the PSO and multi-objective PSO algorithm is provided, alterations to the basic PSO algorithm for the single and multi objective cases are introduced and general notes on GA-PSO comparison are made. In Section III the network design problems are formulated (as single and multi-objective problems) and the corresponding fitness functions are defined. In Section IV, the obtained results of the proposed methodology are critically evaluated against the evolutionary algorithms' performance in terms of design goals satisfaction and convergence behavior, while in Section V the most important conclusions and future work are briefly discussed.

## I. Particle Swarm Optimization

### A. Hybrid Simple PSO

Particle Swarm Optimization (PSO) is a population based algorithm that exploits a set of potential solutions to the optimization problem. Each potential solution is called a particle and their aggregation, in each iteration step, forms the swarm. Swarm particles fly through the multi-dimensional problem space subject to both deterministic and stochastic update rules to new positions, which are subsequently scored by a fitness function. Each particle knows the best position $\vec{p}_l$ that it has ever found, called the local best and is also aware of the best position $\vec{p}_g$ found by any neighbor, called the global best.

$$\begin{aligned} \upsilon_{k+1} &= \upsilon_k + c_1\rho_1\left(p_l - x_k\right) + c_2\rho_2\left(p_g - x_k\right) \\ \upsilon_{k+1} &= 7 sign\left(\upsilon_{k+1}\right)\ \min\left\{\left|\upsilon_{k+1}\right|, \upsilon_{\max}\right\} \qquad (1) \\ x_{k+1} &= x_k + \upsilon_{k+1} \end{aligned}$$

Consequently the individual particles are drawn stochastically toward the positions of their own previous best performance and the best previous performance of their neighbors, in accordance to velocity update equations (1) that should be applied to each dimensional component of the velocity vectors, where *k* denotes the generation number, $\vec{x}_k$ and $\vec{\upsilon}_k$ represent the particle's position and velocity, and $\rho_1$, $\rho_2$ are uniformly distributed random numbers between 0 and 1. The parameter $c_1$, associates the particle's own experience with its current position and is called individuality. The parameter $c_2$ is associated with social interaction between the particles of the neighborhood and is called sociality. The velocity clamping parameter $\upsilon_{max}$ controls the algorithm's step size and is applied to all dimensional components of the particle's velocity, thereby limiting the chances that particles will leave the boundaries of the search space. A more refined way of constraining particles is by using "hard boundary conditions" [12], such as the reflective or absorbing boundary conditions. These do not enforce velocity clipping but rather inverse (RBC) or nullify (ABC) the velocity vector.

When the whole swarm is considered as a neighborhood, the global variant of the PSO is employed. Local neighborhoods may also be used to avoid entrapment of the algorithm on local minima but at the cost of slower convergence. For the network design problem specified in the following section, the Inertia Weight variance of PSO was used [13]. A global as well as a grid neighborhood was defined for improved convergence results and the RBC was enforced along with small $\upsilon_{max}$ values.

A major problem with PSO is premature convergence, which results in great performance loss and sub-optimal solutions. In order to avoid premature convergence, diversity guided PSO was utilized (Attractive and Repulsive PSO [14]) whereas mutation was also applied to the particles—hence the term hybrid simple PSO. ARPSO evaluates the global diversity of the swarm, triggering modes of global attraction or repulsion when predefined thresholds are crossed. Mutation has been demonstrated to successfully complement PSO and improve its performance [15][16][17]. In this study uniform mutation is applied to each particle prior to fitness evaluation with a probability of $p_m$, alternating a predefined percentage of the particle. The application of the mutation operator on the swarm improved significantly the convergence behavior of the algorithm.

### B. Comparison to Genetic Algorithms

The PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms. Both techniques begin with a group of a randomly generated population; utilize a fitness value to evaluate the population and search for the optimum in a partially stochastic manner by updating generations. There are however important differences. Although PSO algorithm retains the conceptual simplicity of the GAs, its' evolutionary process does not create new population members from parent ones; all swarm individuals survive. The original PSO does not employ genetic operators such as crossover and mutation and particles only evolve through their social behavior. Particles' velocities are adjusted, while evolutionary individuals' positions are acted upon. In GAs chromosomes share information with each other, thus the whole population moves like one group towards an optimal area whereas in PSO the information exchange information is a one-way process since only the (local) global best provides information to members of the (sub) swarm. PSO in comparison with GAs, has less complicated operations and it is much easier to implement and apply to design problems with continuous parameters. Moreover in [18] it is shown that binary PSO outperforms GAs in terms of convergence, results and scalability of the problems at hand. In the same study it is suggested that a hybrid model combining characteristics of GA and PSO is preferable. In [19] PSO is shown to exhibit faster convergence compared to GAs. In this paper, the performance of the two algorithms will be compared, for network design problems, since GAs is the EC technique that has been mostly applied to such problems [14].

### *C.Multi Objective PSO*

The main approaches to multi objective optimization consist of weighted aggregation multi objective schemes and population Pareto-based techniques. Pareto-based techniques maintain the set of Pareto-optimal solutions. Pareto-optimal solutions are non dominated solutions in the sense that there are no other superior solutions given the particular search space and set of objectives. Using mathematical notation, a multi-objective optimization problem given a set of *n* objectives and *m* decision parameters is denoted as:

$$\text{Optimize } F(\vec{x})=(f(x), f(x), \ldots, f(x)) \quad (2)$$

Where $\vec{x} \in S$ is a vector satisfying a set of inequality constraints $g_i(\vec{x}) \leq 0,\ i=0,\ldots,k$ . A vector $\vec{u}=(u_{1,}u_{2,}\ldots u_m)$ is said to dominate $\vec{v}=(v_{1,}v_{2,}\ldots v_m)$ if the solution $\vec{u}$ is no worse than $\vec{v}$ in all objectives $(\forall i \in \{1,\ldots,m\},\ F(u_i) \leq F(v_i))$ and the solution vector $\vec{u}$ is better than $\vec{v}$ in at least one objective $(\exists i \in \{1,\ldots,m\},\ F(u_i) < F(v_i))$ . A solution $\vec{x} \in \Omega$ is said to be Pareto-optimal with respect to $\Omega$ , if there is no $\vec{z} \in \Omega$ for which $F(\vec{z})$ dominates $F(\vec{x})$ .

During the past decade, several Multi-Objective Particle Swarm Optimization (MOPSO) methods have been proposed. However there are inherent drawbacks of the PSO algorithm concerning its application on multi-objective optimization. The tendency of the particles to converge to the (single) best solution in the global variant of the PSO is inappropriate for multi-objective optimization, whereas the local variant provides just a refinement near the local optima [20]. Weighted aggregation multi objective schemes were described in [21][22] and Pareto-ranking techniques [23][24][25]. In this study the MOPSO proposed in [25] is used. The particular technique is inspired from MOEA; therefore an external fixed repository is used in which every particle deposits its flight experiences after each flight cycle. The updates to the repository are performed considering a geographically-based system defined in terms of the objective function values of each particle. The search space is divided in hyper-cubes that are appointed a fitness value based on the containing number of particles (fitness sharing). Roulette-wheel is applied to select the hypercube from which a leader for a particle of the swarm will be selected randomly. Mutation as in (II-A) is also applied to the particles with possibility $p_m$.

## II. Real World Telecommunication Tree Network Design Problem

Optimum Communication Spanning Tree Problem (OCSTP) is a special case of the Network Design Problem. Given a connected graph $G=(V,E)$ , *V* represents the node set and *E* the arc set (links). There are communication demands among the nodes V. The traffic demands are specified by a demand matrix *R= (r* $_{ij}$*)* where $r_{ij}$ is the volume of traffic demand among nodes *i, j* for all $(i,j) \in E$ . A distance matrix *D=(d*$_{ij}$*)* represents the distance among sites for all $(i,j) \in E$ . The problem can be stated as: given a set of node locations $G=(V,E)$ specific traffic demand and distance among every set of nodes *i* and *j* , the goal is to construct a spanning tree such that the total cost of communication is minimum among all the spanning trees of *G*. In the classical OCST problem as proposed by Hu [29] the cost of the tree is calculated as the product of the distance of the edge times the overall traffic over the edge. However in real world network design problems, capacity of links is assigned in discrete increments and this is the case that will be addressed in the particular study.

Simple generational genetic algorithms have been proved to provide adequate solutions for the particular problem [10] . However PSO is a simple and robust to control parameters algorithm, whereas the computational efficiency of the technique in comparison the GAs, renders it attractive for application in large scale network design optimization problems. In [26] the authors prove that PSO outperforms GAs in terms of computational efficiency, although the quality of the solutions found is similar in unconstrained nonlinear problems with continuous design variables. Therefore we utilize NetKeys encoding the importance of the links in a continuous manner (see section I ).

In order to evaluate the performance of the PSO on this special case of the network design problem, benchmarking of the algorithm against an instance of the OCST has been performed. Several instances of the OCSTP problem have been presented in literature. In most of them [26][7] the communication cost was given by (3) for a variety of node sets, traffic and distance matrices.

$$min \sum_{i,j \in V} r_{ij} \times d_{p_{(i,j)}} \quad (3)$$

Rothlauf [5] introduced a set of real world telecommunication tree network design problems from a company with nodes located around Germany with modular link capacities. The available capacities for the links are discrete, determined by a fixed cost – for link installation - and a variable cost. The cost of the link per capacity is piecewise linear and monotonically increasing with the length of the link, with decreasing slope (Figure 1).

One instance proposed will be utilised in the particular study. It involves the creation of a communication network including one headquarter and 15 branch offices [10]. It deals with the cost optimization of a rooted spanning tree, where the root of the initial connected graph *G* is the “headquarter” realizing demands towards branch offices. This instance is being studied since it resembles tree/star hierarchical access networks [3]. The demand matrix and the location of the nodes as well as the modular cost function are provided in [28].

The formulation of the single objective problem is given by (4) where F denotes the set of used links, $d_{ij}$ the distance weights of the links between nodes *i* and *j*, $Cap_{i,j}$ the capacity of the links and $b_{ij}$ the traffic flowing directly and indirectly over the link between nodes *i* and *j.* The second equation in (4) denotes that the capacity of the link connecting nodes *i* and *j* must exceed the overall traffic flowing over that link. The form of one available capacity type function is presented in Figure 1.

$$\begin{gathered} min \sum_{i,j \in F} f\left(d_{i,j}, Cap_{i,j}\right) \\ b_{i,j} < Cap_{i,j} \end{gathered} \tag{4}$$

Very often topological design of WANs involves determining the links between nodes given the mean or peak inter node traffic so as to optimize certain QoS parameters [3]. In the multi-objective optimization case presented, the total network cost and average link delay is minimized simultaneously to obtain a Pareto front including optimal non-dominated solutions. A typical *M/M/1* queuing delay model is assumed [1][7] for one type of network service. Therefore two objective functions are used; the cost function delineated by the previous paragraph and the following delay fitness function.

$$min\ AvgDelay$$

$$AvgDelay = \frac{\sum_{i}\sum_{j} \frac{LinkFlow_{i,j}}{Cap_{i,j}} - LinkFlow_{i,j}}{\sum_{i}\sum_{j} LinkFlow_{i,j}} \tag{5}$$

## III. Results

A global and a local version of the hybrid PSO were applied to the Single Objective (SO) problem regarding optimization of network topology deployment. The presented results were obtained after repeating the optimization process for 100 times in order to obtain more statistically important indicators on the algorithm's convergence behavior over several runs.

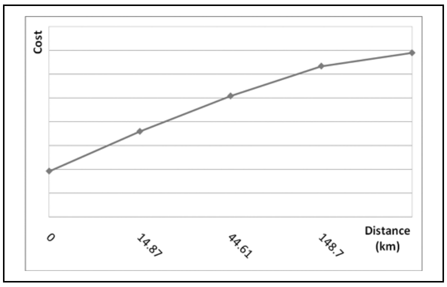


Fig. 1 Link Cost per Capacity

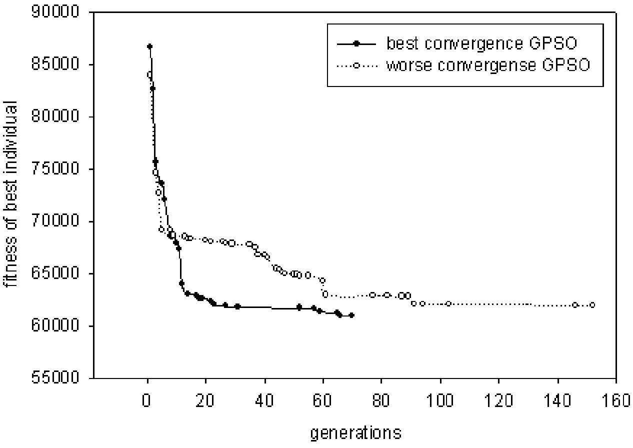


Fig. 2 Convergence Behaviour of Global PSO

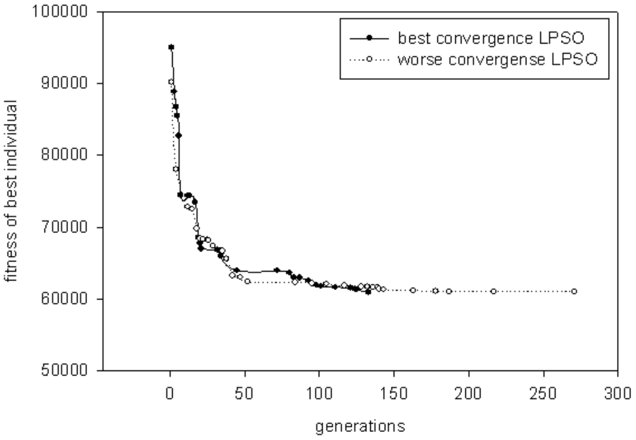


Fig. 3 Convergence Behaviour of Local PSO

The inertia weight PSO was set to sociality $c_1$=0.9, individuality $c_2$=0.9 and a velocity clamping of $\upsilon_{max}$= 0.11. The inertia weight parameter was gradually reduced from $w_1$=0.7 to $w_2$=0.2 during the optimization run. Repulsion threshold was set to 0.1 whereas attraction threshold to 0.2. Mutation was enforced with a probability of 0.005 alternating 10% of the particle (12 edges of the complete graph represented by a particle of 120 edges for 16 nodes). The PSO parameters were found to produce optimal results for the given problem formulation after several trial runs.

In Figure 2 the convergence behavior of the PSO with a global neighborhood is presented while solving the SO network design problem. Indicators included in Table I represent obtained mean fitness value, standard deviation of fitness values and normalized deviation of the mean value from the best solution which is depicted in Figure 4. The best run was able to locate the global best configuration of the network within 70 generations, requiring a total of 17920 fitness function evaluations (for a swarm population of 256 particles). In terms of convergence the worse run was able to obtain a fitness value of 61934.27 on the 152th generation and afterwards stagnated on this local minimum position. Overly, when the algorithm was allowed to evolve for a maximum of 390 generations, it failed to obtain the global minimum 46 times thus yielding an effective success rate of 54%. Although the mean fitness value obtained for the 100 runs (61051.91) is numerically close to the absolute minimum value of 60884.2 for the presented problem, the above results indicate that the use of a global neighborhood can often lead to premature convergence on non-globally optimal solutions.

Respectively, the convergence behavior of the PSO when a rectangular grid neighborhood topology (lateral size = 16) is employed while solving the same network design problem, is presented in Figure 3. The convergence indicators are presented in table Table I (100 runs). The best run was able to locate the global optimum of the problem within 133 generations, requiring a total of 35245 fitness function evaluations (for a swarm population of 265 particles). The worse run was able to obtain a fitness value of 61005.19 on the 271th generation and afterwards stagnated on this local minimum position. When the algorithm was allowed to evolve for a maximum of 390 generations, it failed to obtain the global minimum 11 times thus yielding a success rate of 89%. The obtained mean fitness value (60891.59) improves on the mean value obtained from the PSO with a global neighborhood, indicating that the use of a grid neighborhood improves the convergence behavior of the optimization algorithm for the selected problem. Another key aspect highlighting the above conclusion is that the mean fitness value of the non-successful runs is lower compared to the respective values obtained with a global PSO. Nevertheless, the improved convergence behavior comes at the expense of slower convergence (mean number of generations required to obtain best solution).

The proposed methodology shows an improvement in most cases over the results obtained and presented in [7]. Both PSO variations are able to outperform the performance of the GA with a tournament selection scheme with the exception of the obtained fitness deviation indicator value when the global-neighborhood PSO is used (Table I). The latter can be attributed to the already highlighted worst convergence behavior of the global PSO variant. The indicator values obtained when a grid neighborhood is used shows a ten-fold improvement over the respective GA result for the normalized deviation indicator (0.121% vs. 0.77%).

On the other hand, when a GA is employed using (μ+λ) selection, the obtained mean value is better than the respective value of the grid neighborhood PSO (60886.62 vs. 60891.59), although the PSO optimums have a smaller deviation (40 vs. 23.35). The deterministic nature of the (μ+λ) selection helps to preserve the encoding of the so far best solution found in the population but no analogous mechanism exists in the PSO algorithm. The slightly better performance of the GA over the local PSO in terms of $p_w$ (0.0039 vs. 0.0121%) can be attributed to this effect.

Concerning the multi-criteria instance of the network design problem, the only change applied in PSO parameters was velocity clamping of $\upsilon_{max}$= 0.7. Regarding the multi-objective GA, the algorithm was specified with $p_c$=0.8 and $p_m$=0.005 as crossover and mutation probabilities respectively whereas binary tournament selection, uniform crossover and random mutation on each component of the particle with $p_{m.}$ The results were carried with the same instance of the 16 node network design problem for a population of 265 particles and 400 generations.

The target is to compare the Pareto fronts that are obtained with the same number of fitness evaluations, with a relatively small number of iterations and population size. The fronts from the two multi-objective algorithms are depicted in Figure 5. We notice that the MOPSO algorithm

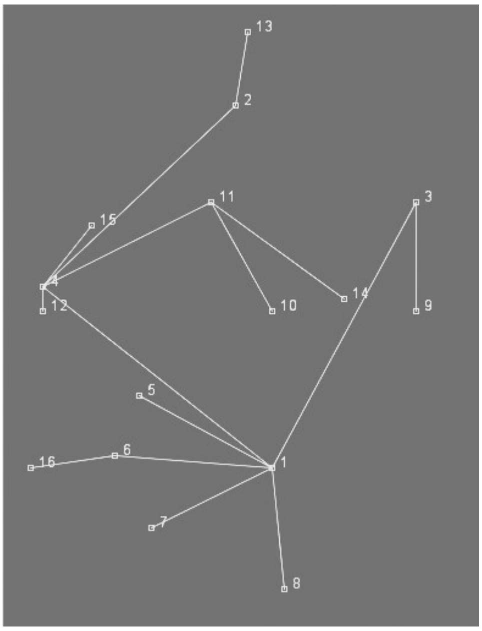


Fig. 4 Optimum Network Topology

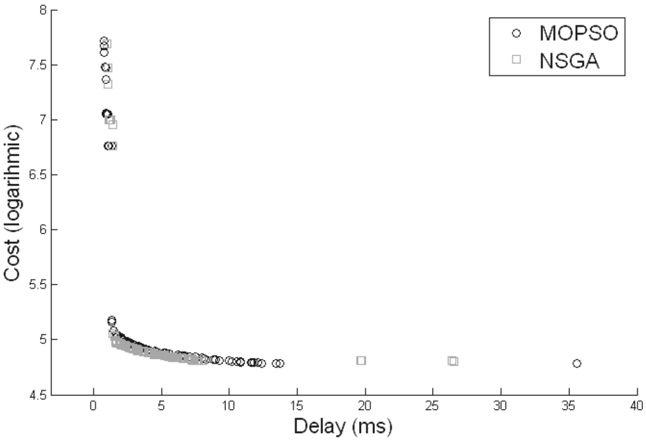


Fig. 5 Pareto Front Comparison among NSGA-II and MOPSO

with mutation, although it acquires a smaller number of non dominated solutions these are more evenly distributed in comparison to NSGA-II. Moreover a wider range of non dominated solutions were obtained in order to approximate the front with more accuracy. With the NSGA-II algorithm a

TABLE I.
COST OF BEST SOLUTION

| Cost of best solution | | Global PSO<br>[Population=256]<br>[Runs=390] | Local PSO<br>[Population=256]<br>[Runs=390] | GA-Tournament<br>[Population=2000]<br>[Runs=30] | GA-(μ+λ)selection<br>[Population=2000]<br>[Runs=50] |
|---|---|---|---|---|---|
| 6883.71 | μ | 61050.91 | 60890.58 | 61353.33 | 60886.62 |
| | σ | 267.77 | 23.35 | 431 | 40 |
| | pw | 0.27% | 0.01% | 0.77% | 0.00% |

region of the front is acquired. In this region it slightly outperforms the MOPSO in terms of the quality of non dominated solutions.

From the network designer's point of view the solution fronts can be divided into three different areas IV, a Low Cost High Delay (LCHD) region, a Medium Cost Medium Delay (MCMD) region and High Cost Low Delay (HCLD) region. In this case in the LCHD area the cost of infrastructure deployment was chosen arbitrarily to be lower than 65.000 whereas in the HCLD area over 150.000 For the LCHD region the MOPSO algorithm with mutation as well as the NSGA-II obtain similar number of solutions though again the range of the MOPSO solutions is greater. Non dominated solutions in this area are easier to obtain than the HCLD region. However it must be noted that concerning the HLDC region the multi-objective PSO outperforms the NSGA-II both in terms of quality and range. Finally the NSGA-II acquires a slightly better approximation of the Pareto front in the MCMD region.

## IV. Conclusions—Future Work

In the particular study, a hybrid version of a single and multi objective PSO algorithm was applied successfully on a communications network topology design problem. The multi-criteria instance of the problem addresses also QoS issues. Optimization objectives in this case are the network layout cost and the average packet delivery delay. NetKeys representation of candidate solutions is utilized. This hybrid version of the particle swarm algorithm applies mutation to the swarm with a probability of $p_m$ whereas additional mechanisms (inertia weight, attraction/repulsion) are utilized to improve convergence behaviour. The proposed methodology shows an improvement in the optimization process in comparison to GAs, concerning both the single and multi objective instance. Future work will include the application of PSO metaheuristic on the topological design of multiservice networks for realistic self similar traffic models.

## References

[1] R. Kumar, N. Banerjee, "Multicriteria network design using evolutionary algorithm", *in 2003 GECCO* , pp. 2179-2190.

[2] M. Pioro, D. Medhi, "Routing, flow and capacity design in communication and computer networks", Morgan-Kaufmann, 2004.

[3] M. Resende, P. Pardalos, "Handbook of Optimization in Telecommunications", Springer, 2006.

[4] P. Kampstra, "Evolutionary Computing in telecommunications, A likely EC success story", Thesis, VU university Amsterdam, The Netherlamds,2005.

[5] F. Rothlauf,, "Towards a theory of representations for genetic and evolutionary algorithms: Development of basic concepts and their application to binary and tree representations", Ph.D. dissertation, Department of Information systems, University of Bayreth, Germany, 2001.

[6] Deb K, Pratap A, Agarwal S, Meyarivan T., "A fast and elitist multiobjective genetic algorithm: NSGA-II.", *in Evolutionary Computation, IEEE Transactions on* , vol.6, no.2, pp.182-197, Apr 2002.

[7] N. Banerjee, R. Kumar, "Multiobjective network design for realistic traffic models.", In *Proceedings of GECCO '07* , pp. 1904-1911, 2007.

[8] A.W. Mohemmed, Z. Mengjie, N.C. Sahoo, "Cooperative Particle Swarm Optimization for the Delay Constrained Least Cost Path Problem", *in Proceedings of EvoCOP 2008,* pp. 25-35, 2008.

[9] A.W. Mohemmed, N.C. Sahoo, T.K. Geok, "A new particle swarm optimization based algorithm for solving shortest-paths tree problems.", *in IEEE CEC 2007*, pp. 3221–3225, 2007.

[10] F. Rothlauf, D. Goldberg, A. Heinzl, "Network random keys—a tree network representation scheme for genetic and evolutionary algorithms", *in Evol. Comput,* vol. 10, no1, pp. 75-97, Mar. 2002.

[11] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *in ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.

[12] S. Mikki, A. Kishk, "Improved particle swarm optimization technique using hard boundary conditions", *in IEEE Microwave and Technology Optical Letters*, vol. 46, no. 5, pp. 422-426, Sep. 2005

[13] R. C. Eberhart, Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* , pp.84-88, 2000

[14] J. Riget, J.S. Vesterstroem, "A diversity-guided particle swarm optimizer—the ARPSO", EVALife Technical Report no. 2002-02.

[15] N. Higashi, H. Iba, "Particle swarm optimization with Gaussian mutation", *in. SIS '03. Proceedings of the 2003 IEEE*, pp. 72-79, April 2003.

[16] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design", *in IEEE Transactions on Systems Man and Cybernetics Part B,* vol.34, no.2, pp. 997-1006, April 2004.

[17] Q. Zhang, X. Li, Q. Trana, "Modified Particle Swarm Optimization Algorithm", *in Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol.5, pp. 2993-2995, Aug. 2005.

[18] R. C. Eberhart, Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", *in Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pp. 611–616, 1998.

[19] R. Hassan, B. K. Cohanim, O. D. Weck, G. A. Venter, "A Comparison of particle swarm optimization and the genetic algorithm". *in Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference*, April 2005.

[20] X. Hu, R. Eberhart, "Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization", *in Proceedings of the 2002 Congress on Evolutionary Computation*, *IEEE World Congress on Computational Intelligence*, pp. 1677-1681, May 2002.

[21] K. E. Parsopoulos, M. N. Vrahatis. "Particle Swarm Optimization Method in Multiobjective Problems", *in Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pp. 603-607, 2002.

[22] U. Baumgartner, Ch. Magele, W. Renhart. "Pareto optimality and particle swarm optimization", *in Magnetics, IEEE Transactions on* , vol. 40, no.2, pp. 1172-1175, March 2004.

[23] J. E Fieldsend, R. M. Everson, S. Singh, "Using unconstrained elite archives for multiobjective optimization", *Evolutionary Computation, IEEE Transactions on* , vol.7, no.3, pp. 305-323, June 2003.

[24] T. Ray, K. M. Liew. "A swarm metaphor for multiobjective design optimization", *in Engineering Optimization*, vol. 34, no. 2, pp. 141–153, March 2002.

[25] C. A. Coello, M. S. Lechunga, "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization", *in Proceedings of the 2002 Congress on Evolutionary Computation*, *IEEE World Congress on Computational Intelligence,* pp. 1051-1056, May 2002.

[26] C. Palmer, "An Approach to a Problem in Network Design Using Genetic Algorithms" Ph.D. dissertation, Polytechnic University, Brooklyn, New York.

[27] G. R. Raidl, "Various instances of optimal communication spanning tree problems," personal communication, February 2001.

[28] Franz Rothlauf, "Representations for genetic and evolutionary algorithms", Springer-Verlang, 2006.

[29] T. C. Hu, "Optimum communication spanning trees", *in SIAM J. on Computing* , p.p. 188-195, 1974.

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Two stages optimization problem: New variant of Bin Packing Problem for decision making

Ahmad Shraideh
LAGIS Ecole Centrale de Lille
Avenue Paul Langevin
59651 Villeneuve d'Ascq
France
Email: ahmad.shraideh@ec-lille.fr

Hervé Camus
LAGIS Ecole Centrale de Lille
Avenue Paul Langevin
59651 Villeneuve d'Ascq
France
Email: herve.camus@ec-lille.fr

Pascal Yim
3Suisses International Group
Cité Numérique,245 rue Jean Jaurès
59650 Villeneuve d'Ascq
France
Email:pyim@3suisses.fr

***Abstract*—In this paper, we present a new multi-criteria assignment problem that groups characteristics from the well known Bin Packing Problem (BPP) and Generalized Assignment Problem (GAP). Similarities and differences between these problems are discussed, and a new variant of BPP is presented. The new variant will be called generalized assignment problem with identified first-use bins (GAPIFB). The GAPIFB will be used to supply decision makers with quantitative and qualitative indicators in order to optimize a business process. An algorithm based on the GAP problem model and on GAPIFB is proposed.**

## I. Introduction

Within the context of the French competitive cluster[1] "Industrie du commerce", with the collaboration of COFIDIS[2] and ALFEA consulting[3], the project GOCD[4] aims to set up a new dematerialized workflow system, to treat the received contracts at COFIDIS. Our participation was to install a new optimization and decision-making tool for the new system with the necessary key performance indicators. Every day, COFIDIS receives from the post office thousands of contracts and credit demands of different types (for facility we will use the terms contracts for contracts and credit demands). The quantities and the types of contracts are known in the morning and can change from one day to another. Some contracts must be treated at the same day others can wait for some days. The treatment time for a contract by a collaborator is defined by a matrix of competence, as each collaborator has different skills and experiences with respect to contract type. The contracts are distributed currently to company collaborators according to past acquired experience in heuristic method. This distribution is not optimal, but hoped to be approximated to the optimal one. The daily work hours for the collaborators are not equal, in reason of human resources management considerations. If the capacity of the collaborators is overloaded, the decision makers can either come to the aid of temporary workers to treat the overloaded contracts or they can holdup the treatment of unimportant type of contracts. The objective is to find the best distribution of contracts that will best exploit collaborators capacities, and to determine if the daily load of contract exceeds current collaborators capacities. In this case decision makers need to know the exact number of temporary workers required to treat all the contracts. This problem can be seen as a multi-criteria optimization problem with discrete variables.

In examining the actual used heuristic, we can identify three problems. The first one concerns the assignment method used to distribute contracts. In this method, contracts are distributed depending on previous experience, and not on approved optimal method. Bad distribution of contracts could lead to unnecessary call of temporary workers. Different distributions of contracts could result in different total time of treatment, this can be seen clearly when the current load of contracts is close to company optimal capacity. The second problem is to detect in advance overloaded situations and to decide which contracts to treat if decision makers don't prefer to hire temporary workers. The third problem is to determine the exact number of temporary workers when needed. To the best of our knowledge, there is no model capable to represent completely these problems; rather we find models with partial solution for partial problem.

The paper is organized as follow. In the next section, a formulation of the problem is presented. In section three, we study two famous assignment problems, the BPP problem and GAP problem. The different variants of each problem are discussed and their weaknesses regarding our problem are clarified. We will demonstrate that neither of these problems can solitary gives a complete answer to the mentioned problem. In section four, we present our approach to solve the problem, followed by mathematical formulation and evaluation results. We terminate by our conclusions and future work.

## II. Problem formulation

A generic formulation and notation for the problem is as the following

- $NC$ : Number of all tasks (contracts),
- $N$ : Number of primary agents (company workers),

[1] A competitive cluster is an initiative that brings together companies, research centers and educational institutions in order to develop synergies and cooperative efforts. http://www.industrie.gouv.fr/poles-competitivite

[2] French consumer credit company. http://www.cofidis.com

[3] French information system consulting company. http://www.alfea-consulting.com

[4] GOCD : French acronym for Management and optimization of document life cycle

978-83-60810-14-9/08/$25.00 © 2008 IEEE

- $M$ : Number of available secondary agents (temporary workers),
- *CAP*: Primary or secondary agents capacities (in hours), $CAP = \{CAP_1, CAP_2, .., CAP_N, .., CAP_{N+M}\}$,
- $T_{ij}$: Needed time for primary or secondary agent *i* to treat task *j*,
- $U_i$: Boolean. 1 if primary or secondary agent *i* is used, otherwise 0, $U = \{U_1, U_2, .., U_N, .., U_{N+M}\}$,
- $X_{ij}$: Boolean. 1 if task *j* is assigned to primary or secondary agent *i*, otherwise 0.

The objective functions are

$$Min \sum_{i=1}^{N+M} U_i \tag{1}$$

$$Min \sum_{j=1}^{NC} X_{ij} \times T_{ij} \tag{2}$$

Subject to,

$$\sum_{i=1}^{N+M} X_{ij} = 1, \qquad \forall j \in \{1, 2, .., NC\} \tag{3}$$

$$\sum_{j=1}^{NC} X_{ij} \times T_{ij} \leq CAP_i \times U_i, \qquad \forall i \in \{1, 2, .., N+M\} \tag{4}$$

$$U_i = 1, \qquad \forall i \in \{1, 2, .., N\} \tag{5}$$

Notice that for all *i* ∈{1, 2,.., N}, $U_i$ represents a primary agent and for all *i* ∈{N+1, N+2,.., N+M}, $U_i$ represents a secondary agent. The objective function (1) searches to minimize the number of secondary agents used to treat all tasks. We have chosen in objectives function (2) to minimize the total treatment time as example, other objectives can be designed by the decision makers. Constraint (3) indicates that all tasks must be distributed, and each task is given only to one agent. Constraint (4) explains that the capacity of each used agent must not be violated. Finally constraint (5) is used to be sure that all primary agents are used. As we can see, our problem is a specific case of this generic case, where the contract treatment time is defined by its type and the agent treating it. For simplification, in our proposed solution, we will reformulate the generic problem and new notations will be used where the contracts of the same type assigned to one agent are presented by one integer variable in stead of a set of binary variables for each contract. In fact, we pass from linear multi-criteria problem with binary integer to linear multi-criteria problem with integer variables and this will reduce the number of variable to use. For example, an instance of the problem with 100 primary agents, 10 available secondary agents, 5 contract type and 3500 contracts, will need 385110 binary variables if we use binary representation (100+10 variable to present agents and 100+10*3500 to present the contracts). Whereas by grouping the contract of the same type assigned to an agent will require 660 variables (100+10 variable to present agents and 100+10*5 to present the assigned contracts)

## III. Bin packing and Generalized assignment problems

In literature, we find some assignment problems which are similar to our problem. These problems were widely studied and analyzed. The closest ones to our problem are Bin Packing Problem and Generalized Assignment problem.

Bin packing problem(BPP) is well known for being one of the combinatorial NP-hard problems [1]. Many researches were realized to find the optimal or an approximated solution for this problem [2] [3] [4]. In its simplest form, we have a set of bins of equal capacity and a list of objects, each object has an equivalent weight (costs of treatment) for all bins. The objective is to find the minimum number of bins in order to pack all the objects in the list. A bin packing problem can be either on-line or off-line. In on-line packing problem, we have information only about the current object in the list to be packed, and no objects can be repacked later. In off-line packing problem complete information about all objects are known in advance. Variants of BPP include, two dimensional bin packing [5] [6] [7] [8] and three dimensional bin packing problem [9] [10], in which each object have either two dimensions (area) or even three dimensions (volume). Another variant and well studied BPP is the extendable bin packing problem [11] [12] , where the sizes of the bins are extendable when necessary to answer work needs.

Another famous problem is the generalized assignment problem (GAP), a generalization of Multi-knapsack problem [13]. In GAP problem, a set of objects with cost and profit, are assigned to a set of agents. Each object can be allocated to any but only one agent, and the treatment of an object needs resources which change, depending on the object and the agent treating it, each agent can have different capacity. The objective is to maximize the profit without exceeding agent's capacities. A survey on the algorithms used to solve this problem can be found in [14].

It is clear that the two models have different objectives and different formulation. In the BPP problem, we search to minimize the number of used bins to pack all the objects without any consideration to profit. Whereas in the GAP problem, profit is considered but the allocation of all objects is not important, which means the possibility to have an optimal solution without distributing all objects. More over, in BPP the objects have an equal value whatever was the bin used to pack them, which is not the case in the GAP problem, where the profit of an object depends on the object and on the agent. From the previous description for BBP and GAP problem, we see that our assignment problem corresponds to GAP problem in that it search to optimize certain predefined objectives by decision makers. But unfortunately, it is unable to determine the minimum number of temporary workers needed to achieve these objectives in overloaded cases, as the number of bins (workers in our problem) in GAP problem must be defined in advance.

On the other hand, BPP model can find the minimum number of workers to treat all the contracts. Still neither classical form nor its variants are capable to distinct between company collaborators and temporary workers in there solution. This can lead to solutions which exclude some company workers if the utilization of temporary workers gives better solutions than using company workers. This is an important matter, as we must first verify company workers capability to treat all the contracts and to wait decision maker to decide whether to hire temporary workers or not. Before integrating temporary workers with company workers to find optimal solution.

## IV. Proposed Approach

To solve this multi-criteria problem, we propose to decompose it into two mono-criteria problems. Each mono-criteria problem is presented by a model, and an exact method is used to find the optimal solution for each of these models as the size of our problem is not large. We can imagine this solution in two stages. In the first stage, we use GAPIFB model to search the minimum number of secondary agents needed to treat all contracts. This is a major concern to enterprise managers as it is considered the most enterprise financial resources consumers. In GAPIFB, a set of tasks (objects) must be assigned to a set of agents (bins). The size of each task can vary from one agent to another, agent's capacities are not equal, and the set of agents includes two types of agents, primary agents and secondary agents. The use of secondary agents is allowed only when the primary agents are not capable to treat all the tasks. The objective function is to find the minimal number of secondary agents to be used with the primary agent to treat all the received contracts. At the end of this stage, decision maker not only knows company situation if it is overloaded or under loaded but also he knows the exact number *EN* of secondary agents needed to treat all contracts in overload situation. Decision maker must also decide whether the company will hire secondary workers or not, and if the decision was to hire secondary workers, is it to hire the exact number of secondary workers needed to treat all the contracts or to hire certain number *L* of needed secondary workers?

In the second stage, a group of objectives function is available to decision maker which supply him with comprehensive vision of all possible decision scenarios that can be taken, and their effect on company contracts distribution process. The choice of this objective is left to decision makers. One objective can be to give high treatment priority to contracts that can not be delayed or to contracts considered as profitable to the company. Another objective could be to treat important contracts types uniquely by company collaborators as they have the best experience and skills. The fairness of collaborators loads can be significance objective from the social vision of point. To maximize the rate of profitability by collaborators can be an interesting objective from economic vision. For facility in this paper, an objective function which is to minimize the total treatment time of contracts was been chosen. Figure 1 demonstrates decision making process for our approach. Mathematical formulation for GAPIFB and GAP models are discussed in details in the next sections.

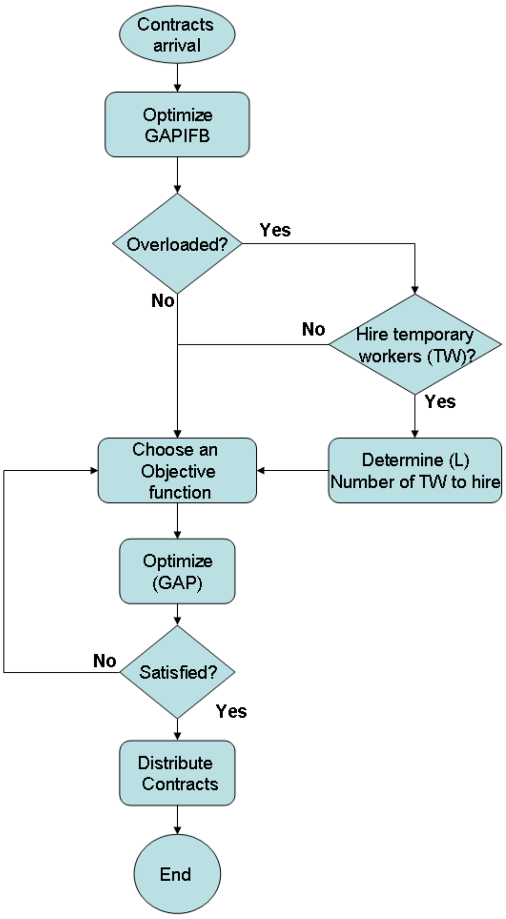


Fig. 1. Proposed decision making process

### A. Mathematical formulation

For the first stage, a linear formulation with integer variables is used to present GAPIFB. In GAPIFB, a set of tasks(contracts) must be assigned to a set of agents(bins). The size of each task can vary from one agent to another, agent's capacities are not equal, and the set of agents includes two types of agents, primary agents and secondary agents. The use of secondary agents is allowed only when the primary agents are not capable to treat all the tasks. The objective function is to minimize the number of secondary agents used with primary agents to treat the whole quantity of received tasks. In this formulation we used binary variable $U_i$ to represent both primary agents and secondary agents, where for *i* ∈{1, 2,.., N}, $U_i$ represents a primary agent and for *i* ∈{N+1, N+2,.., N+M}, $U_i$ represents a secondary agent. $X_{ij}$ is used to indicate the number of tasks of type *j* attributed to primary or secondary agent *i*. Notice that in this stage, $X_{ij}$ is an integer variable and not binary variable, and $T_{ij}$ presents the treatment time for contract of type *j* by agent *i*. The new modified notations and a formulation to the first stage objective function with constraints are given as follows:

- *Z*: Number of tasks types,
- $QT_j$: Quantity of tasks of type *j*,

- $T_{ij}$: Needed time for primary or secondary agent *i* to treat a task of type *j*,
- $X_{ij}$: The number of tasks of type *j* assigned to primary or secondary agent *i*.

$$Min \sum_{i=1}^{N+M} U_i \tag{6}$$

Subject to

$$\sum_{j=1}^{Z} X_{ij} \times T_{i,j} \leq CAP_i \times U_i, \qquad \forall i \in \{1,2,..,N+M\} \tag{7}$$

$$\sum_{i=1}^{N+M} X_{ij} = QT_j, \qquad \forall j \in \{1,2,..,Z\} \tag{8}$$

$$\sum_{i=1}^{N} U_i = N, \tag{9}$$

Constraint (7) ensures that the capacity of agents is not violated. Constraint (8) ensures that all tasks are allocated and each task is assigned to only one agent. To ensure that the solver will search the optimal solution within the solutions that used all primary agents we added constraint (9), where $U_i$ presents a primary worker ∀ i ∈{1,2,...,N}. Without constraint (9), it is possible that the solver gives us a solution that excludes some primary agent if the use of secondary agents gives better results.

The second stage is formulated as classical GAP problem. Consider the following.

- *L* : Number of secondary agents to be hired (decision maker decision),
- $i \in \{1, 2, ..., L\}$ and $j \in \{1, 2, ..., Z\}$.

As we mention before we consider here only one objective function which is to minimise the total treatment time for all contracts. In using the new notation, the objective function (2) is replaced by the objective function (10) as the following:

$$Min \sum_{i=1}^{N+L} \sum_{j=1}^{Z} X_{ij} * T_{ij} \tag{10}$$

Subject to

$$\sum_{j=1}^{Z} X_{ij} \times T_{ij} \leq CAP_i, \qquad \forall i \in \{1,2,..,L\} \tag{11}$$

$$\sum_{i=1}^{L} X_{ij} = QT_j, \qquad \forall j \in \{1,2,..,Z\} \tag{12}$$

Constraint (11) ensures agent capacity not to be violated. Constraint (12) ensure that the distributed quantity of tasks type *k* is less than the received quantity of that type.

### B. *Evaluation and test results*

In order to evaluate our approach, a formulation of the problem as mixed integer program was realized[5]. The optimal solution in the two stages were computed using Cplex9[6] solver, which uses an advanced mathematical programming and constraint-based optimization techniques. Many samples were generated randomly with different numbers of primary agents and secondary agents and with different competence matrix and different capacities for each sample. The quantities and types of each tasks was generated to be near to the average of company capacity. It was found that the proposed approach is capable to detect under loaded situation and to find the optimal solution to distribute all tasks. In overloaded situation, the proposed GAPIFB model was able to find the minimum number of secondary agents needed to treat all the tasks, all company agents were included in every produced solution. Within the size of our problem, the execution time was ordered in milliseconds for bothe the first stage and in the second stage. This execution time was very satisfactory for company decision makers. When the number of contracts increases respectively with the number of agents (primary and secondary), execution time is increased to be in seconds. In fixing the number of primary agent and increasing the number of available secondary agent the execution time increase considerably. Table 1 shows the execution time for some realized samples.

## V. Conclusion

In this paper, we presented a new multi-criteria assignment problem for decision making and proposed a new exact approach to solve it. The problem consists of allocating a set of different type of tasks to a set of primary agents; in case of overload secondary agents can be used to treat all tasks. Each agent has different capacity and different experience per task type according to matrix of competence. The treatment time of a task, as a result, will depend on the type of the task and the agent treating it. The first objective is to determine company situation (underloaded or overloaded) and to give the exact number of secondary agent in the overloaded cases. The second objective is to give a comprehensive vision of all possible scenarios for allocating tasks to decision maker. To solve this problem we divided it into two parts (stages) with mono-objective function for each. Exact methods were used to solve each stage. In the first part, we used the proposed GAPIFB. This model is able to distinguish between primary agent and secondary. Primary agents imperatively appear in the optimal solution, which is not the case in using simple form of BPP where the solver search the optimal solution whatever was the agent to use. This makes it possible to exclude some primary agents if the use of secondary agent gives best solution. In addition, simple BPP defines fixed treatment cost by task type, which is not the case in GAPIFB.

[5] Tests were held on Intel Dual Core T7200 2.00GHz machine, 2Go of RAM

[6] CPLEX : an optimization software package produced by ILOG. http://www.ilog.com/

TABLE I
SIMULATION RESULT FOR THE FIRST STAGE—GAPIFB OPTIMISATION

| Primary agents | Secondary agents | Number of contracts | Execution time second |
|---|---|---|---|
| 100 | 15 | 3500 | 0.047 |
| 100 | 15 | 3500 | 0.031 |
| 100 | 15 | 3500 | 0.031 |
| 500 | 75 | 17500 | 0.078 |
| 500 | 75 | 17500 | 0.078 |
| 500 | 75 | 17500 | 0.073 |
| 1000 | 150 | 35000 | 0.141 |
| 1000 | 150 | 35000 | 0.125 |
| 1000 | 150 | 35000 | 0.14 |
| 5000 | 750 | 175000 | 1.00 |
| 5000 | 750 | 175000 | 1.031 |
| 5000 | 750 | 175000 | 1.016 |
| 10000 | 1500 | 350000 | 3.172 |
| 10000 | 1500 | 350000 | 3.14 |
| 10000 | 1500 | 350000 | 3.187 |
| 15000 | 2250 | 525000 | 6.375 |
| 15000 | 2250 | 525000 | 6.312 |
| 15000 | 2250 | 525000 | 6.297 |

In the second stage, a GAP model is used with different objective functions to give decision makers ideas about the suitable method to distribute the tasks. Others objectives can be used, this choice is left to company decision makers.

A formulation with integer variables was used instead of binary variable to implement GABIFB, which reduce the total number of variables. In this formulation, both primary and secondary agents are represented as binary variables. Many samples were generated and tested, and our approach proved to be capable to define the minimum number of needed secondary agent. The time of execution was ordered in milliseconds. Future work can be conceived to extend our work in order to deal with and take in consideration contracts flow for long period e.g. one week flow. Other objective functions to distribute the contracts can be imagined in order to construct and supply new efficient Key Performance Indicators (KPI) for the decision makers.

## REFERENCES

[1] D. J. M Garey, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.

[2] D. J. Brown, “A lower bound for on-line one-dimensional bin packing algorithms,” Illinois Univ at Urbana-Champaign applied computation theory group, Tech. Rep., 1979.

[3] J. Csirik and G. J. Woeginger, *On-line packing and covering problems*. Springer Berlin / Heidelberg, 1998, ch. 7, pp. 147 – 177.

[4] L. Epstein and R. van Stee, “Online bin packing with resource augmentation,” *Discrete Optimization*, vol. 4, pp. 322–333, 2007.

[5] F. R. K. Chung, M. R. Garey, and D. S. Johnson, “On packing two-dimensional bins,” *SIAM Journal of Algebraic and Discrete Methods*, vol. 3, pp. 66–76, 1982.

[6] J. O. Berkey and P. Y. Wang, “Two-dimensional finite bin-packing algorithms,” *The Journal of the Operational Research Society*, vol. 38, pp. 423–429, 1987.

[7] A. Lodi, S. Martello, and M. Monaci, “Two-dimensional packing problems: A survey,” *European Journal of Operational Research*, vol. 141, pp. 241–252, 2002.

[8] J. Puchinger and G. R. Raidla, “Models and algorithms for three-stage two-dimensional bin packing,” *European Journal of Operational Research*, vol. 183, pp. 1304–1327, 2007.

[9] S. Martello, D. Pisinger, and D. Vigo, “The three-dimensional bin packing problem,” *OPERATIONS RESEARCH*, vol. 48, pp. 256–267, 2000.

[10] F. Miyazawa and Y. Wakabayashi, “Two- and three-dimensional parametric packing,” *Computers & Operations Research*, vol. 34, pp. 2589–2603, 2007.

[11] P. Dell’Olmo, H. Kellerer, M. G. Speranzac, and Z. Tuza, “A 13/12 approximation algorithm for bin packing with extendable bins,” *Information Processing Letters*, vol. 65, pp. 229–233, 1998.

[12] E. Coffman and G. S. Lueker, “Approximation algorithms for extensible bin packing,” *Journal of Scheduling*, vol. 9, pp. 63–69, 2006.

[13] Silvano and P. T. Martello, *Knapsack problem algorithms and computer implementation*. John Wiley and Sons, 1990.

[14] D. G. Cattrysse and L. N. V. Wassenhove, “A survey of algorithms for the generalized assignment problem,” *European Journal of Operational Research*, vol. 60, pp. 260–272, 1992.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 927–931

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Adaptive Differential Evolution and Exponential Crossover

Josef Tvrdík
University of Ostrava, Department of Computer Science
Email: josef.tvrdik@osu.cz

***Abstract*—Several adaptive variants of differential evolution are described and compared in two sets of benchmark problems. The influence of exponential crossover on efficiency of the search is studied. The use of both types of crossover together makes the algorithms more robust. Such algorithms are convenient for the real-world problems, where we need an adaptive algorithm applicable without time-wasting parameter tuning.**

## I. Introduction

THE GLOBAL optimization problem is considered in this paper in the following form:

$$\text{minimize } f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{x} \in D,$$

where $\boldsymbol{x}$ is a continuous variable with the domain $D \subset \mathbb{R}^d$, and $f(\boldsymbol{x}) : D \rightarrow \mathbb{R}$ is a continuous function. The domain $D$ is defined by specifying boundary constrains, that are lower $(a_j)$ and upper $(b_j)$ limits of each component $j$, $D = \prod_{j=1}^{d}[a_j, b_j],\ a_j < b_j,\ j = 1, 2, \dots, d$. The global minimum point $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in D} f(\boldsymbol{x})$ is the solution of the problem. The global optimization problems that use boundary constrains only are usually referred as unconstrained ones [1], due to the fact, that boundary constrains are easy to handle and in computer numerical solution there are ”natural” boundary constrains, given by the representation of numbers in floating-point format.

The global optimization problem is not easy to solve and standard deterministic algorithms tend to stop the search in local minimum nearest to the input starting point. Therefore, heuristic search is widely used in the global optimization. Such heuristics are often inspired by natural processes, mainly by the evolution in populations. Evolutionary algorithms are able to find the acceptable solution with reasonable time demand, but efficiency of the search is sensitive to the setting of control parameters. Application of evolutionary algorithms usually requires time-consuming tuning of control parameters to the problem to be solved. Thus, great effort has been focused to the development of self-adaptive variants of evolutionary algorithms, applicable without preliminary tuning of control parameter.

## II. Differential Evolution

The differential evolution (DE) was introduced by Storn and Price [2]. Nowadays the DE has become one of the most frequently used evolutionary algorithms solving the global optimization problems [3]. The algorithm of DE in pseudo-code is shown in Fig. 1. New trial point $\boldsymbol{y}$ (line 4 in DE algorithm) is generated by using mutation and crossover.

There are various strategies, how to create the mutant point $\boldsymbol{v}$. The most popular strategy denoted by abbreviation DE/rand/1/ generates the point $\boldsymbol{v}$ by adding the weighted difference of two points

$$\boldsymbol{v} = \boldsymbol{r}_1 + F\left(\boldsymbol{r}_2 - \boldsymbol{r}_3\right), \tag{1}$$

where $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{r}_3$ are three mutually distinct points taken randomly from population $P$, not coinciding with the current $\boldsymbol{x}_i$, and $F > 0$ is input parameter.

```
1 initialize population $P = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N)$, $\boldsymbol{x}_i \in D$
2 repeat
3     for $i := 1$ to $N$ do
4         generate a new trial vector $\boldsymbol{y}$
5         if $f(\boldsymbol{y}) < f(\boldsymbol{x}_i)$ then insert $\boldsymbol{y}$ into new generation $Q$
6                             else insert $\boldsymbol{x}_i$ into new generation $Q$
7         endif
8     endfor
9     $P := Q$
10 until stopping condition
```

Fig. 1. Differential evolution—Algorithm in pseudo code

Kaelo and Ali [4] proposed a slightly different attempt to generating a mutant point $\boldsymbol{v}$ by (1). They called it random localization. The point $\boldsymbol{r}_1$ is not chosen randomly, but it is tournament best among $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$, that is $\boldsymbol{r}_1 = \arg\min_{i\in\{1,2,3\}} f(\boldsymbol{r}_i)$.

The crossover operator constructs the offspring $\boldsymbol{y}$ by mixing components of the current individual $\boldsymbol{x}_i$ and the point $\boldsymbol{v}$ generated by mutation. There are two types of crossover used in DE, binomial and exponential ones.

Binomial crossover replaces the elements of vector $\boldsymbol{x}_i$ using the following rule

$$y_j = \begin{cases} v_j & \text{if} \quad U_j \le CR \quad \text{or} \quad j = l \\ x_{ij} & \text{if} \quad U_j > CR \quad \text{and} \quad j \ne l\,, \end{cases} \tag{2}$$

where $l$ is a randomly chosen integer from $\{1, 2, \dots, d\}$, and $U_1, U_2, \dots, U_d$ are independent random variables uniformly distributed in $[0,\ 1)$. $CR \in [0, 1]$ is an input parameter influencing the number of elements to be exchanged by crossover. Eq. (2) ensures that at least one element of $\boldsymbol{x}_i$ is changed, even if $CR = 0$. This kind of crossover according to (2) is commonly expressed by abbreviation DE/·/·/bin.

For exponential crossover (DE/·/·/exp), the starting position of crossover is chosen randomly from $1, \ldots, d$, and $L$ consecutive elements (counted in circular manner) are taken from the mutant vector $\boldsymbol{v}$. Probability of replacing the $k$-th element in the sequence $1, 2, \ldots, L, \quad L \leq d$, decreases exponentially with increasing $k$. There are two main differences between binomial and exponential crossovers:

- $L$ adjacent elements are changed in exponential variant, in binomial one the changed coordinates are dispersed randomly over the dimensions $1, 2, \ldots, d$.
- While the relation between the probability of crossover and the *CR* is linear in binomial crossover; in the exponential one, this relation is nonlinear, and the deviation from linearity enlarges with increasing dimension of problem.

Probability of crossover, $p_m$, determines the number of exchanged elements ($p_m$ being the mean value of relative frequency). Zaharie [5] derived the relation between $p_m$ and *CR* for exponential crossover. Her result can be rewritten in the form of polynom

$$CR^d - \ d\, p_m\, CR + \ d\, p_m - 1 = 0. \tag{3}$$

It is apparent that the polynom (3) has the only one real root in the open interval of $(0, 1)$ for $p_m \in (1/d, 1)$. The crossover parameter $CR = 0$ for $p_m = 1/d$, and for $p_m = 1$ the value of $CR = 1$. Thus, for given $p_m$ we can find unique corresponding value of *CR*, to be set as crossover parameter.

Differential evolution has a few control parameters only, namely the size of population $N$, selection of mutation strategy, choice of crossover type, and pair of parameters $F$ and *CR*. However, the efficiency of differential evolution is very sensitive especially regarding the setting of $F$ and *CR* values. The most suitable control parameters values for specific problem may be found by trial-and-error tuning, but it requires a lot of time. There are some recommendations for the setting of these parameters. Zaharie [6] suggested the intervals for the control parameters based on the variability of population, for other proposals see [2], [3], [7], [8].

Self-adaptive setting of control parameters was studied in several papers. Ali and Törn [9] proposed a simple rule for adapting the $F$ scaling factor value during the search process. Some other attempts to the adaptation of DE control parameters are summarized in Liu and Lampinen [10]. Recently, Quin and Suganthan [11] proposed self-adaptive choice of mutation strategy combined with controlled random adjusting the values of $F$ and *CR*.

Evolutionary self-adaptation of control parameters $F$ and *CR* suggested by Brest et al. [12] and competitive self-adaptive setting of the control parameters introduced by Tvrdík [13] (both variants with binomial crossover) have proved good convergence in the applications to various problems of the global optimization [14]. Experimental comparison of these self-adaptive patterns and influence of exponential crossover is the goal of this paper. These self-adaptive algorithms are described in detail in next section.

## III. Variants of DE in Comparison

DE with evolutionary self-adaptation of $F$ and *CR* was proposed by Brest et al. [12]. The values of $F$ and *CR* are initialized randomly for each point in population and survive with the individuals in population, but they can be mutated randomly in each generation with given probabilities $\tau_1,\ \tau_2$. Authors used binomial crossover with the values of $CR \in [0,\ 1]$ distributed uniformly, and DE/rand/1 mutation with values of $F$ also distributed uniformly in $[\,F_l,\ F_u]$. They set input parameters $F_l = 0.1$, and $F_u = 0.9$.

The competitive setting of the control parameters was described in [13]. In this self-adaptation approach, we choose among $H$ different settings of $F$ and *CR* randomly with probabilities $q_h,\ h = 1, 2, \ldots, H$. These probabilities are modified according to the success rate of the settings in preceding steps of search process. The $h$-th setting is considered successful, if it generates such a trial point $\boldsymbol{y}$ that $f(\boldsymbol{y}) < f(\boldsymbol{x}_i)$. Probability $q_h$ is evaluated as the relative frequency

$$q_h = \frac{n_h + n_0}{\sum_{j=1}^{H}(n_j + n_0)}\ , \tag{4}$$

where $n_h$ is the current count of the $h$-th setting successes, and $n_0 > 0$ is a constant. The setting of $n_0 > 1$ prevents a dramatic change in $q_h$ by one random successful use of the $h$-th parameter setting. To avoid degeneration of the search process, the current values of $q_h$ are reset to their starting values $q_h = 1/H$, if any probability $q_h$ decreases bellow some given limit $\delta > 0$. Several variants of such competitive differential evolution with binomial crossover were numerically compared in [15], and two best performing variants were included into open Matlab library [16].

The mutation according to (1) could cause that a new trial point $\boldsymbol{y}$ moves out of the domain $D$. In such case, the point $\boldsymbol{y}$ can be skipped and a new one generated. More effective attempt is to replace value of $y_j < a_j$ or $y_j > b_j$, either by random value in $[a_j,\, b_j]$, see [7], or by reversed value of this coordinate $y_j$ into $D$ over the $a_j$ or $b_j$ by mirroring, see [17]. The latter method is used in algorithms implemented for numerical tests.

The goal of experiments was to compare the impact of crossover type to the performance of in the both self-adaptive kinds of DE algorithm.

The four most efficient competitive variants from twelve variants of DE tested in [18] are selected for experimental comparison. These variants have proved significantly better efficiency than the standard DE. Two variants in experiments are based on the Brest approach to self-adaptation [12] and two other variants in experimental tests combine both approaches. The tested variants are denoted by mnemonic labels.

The labels of competitive variants begin with capital letter ”C” in bellow text. In variant label, the type of crossover is followed by number of competing settings. The presence of random localization is marked by suffix ”rl”. Where the variant uses both types of crossover, ”rl” is related to both of them. Nine settings of $F$ and *CR* for binomial crossover are created from all combinations of $F \in \{0.5, 0.8, 1\}$ and *CR* $\in \{0, 0.5, 1\}$). If there are 6 settings in competition only, the value $F = 1$ is not used. The same values of $F$ are used for exponential crossover. However, *CR* values are derived from the crossover probability, given by (3). Actual values of crossover probability can vary in interval $[1/d,\ 1]$, because at least one coordinate of current point $\boldsymbol{x}_i$ in changed in both types of crossover. Notice, that for extreme values of probability, $p_m = 1/d$ and $p_m = 1$, exponential crossover does not differ from binomial one. In order to ensure the different behaviour of exponential crossover from binomial one, the following three levels of crossover probability $p_1$, $p_2$, $p_3$ were chosen. Value of $p_2$ is in the middle of the interval $[1/d,\ 1]$, $p_1$ in the middle of the interval $[1/d,\ p_2]$, and $p_3$ in the middle of the interval $[p_2,\ 1]$, i.e.

$$p_2 = \frac{1 + 1/d}{2}, \quad p_1 = \frac{1/d + p_2}{2}, \quad p_3 = \frac{p_2 + 1}{2}. \tag{5}$$

The values of crossover probability and the corresponding values of parameter *CR* for $d = 10$ and $d = 30$ are given in Table I.

TABLE I
VALUES OF CROSSOVER PROBABILITY AND THE CORRESPONDING VALUES OF *CR* PARAMETER FOR EXPONENTIAL CROSSOVER

| | $d = 10$ | | | | $d = 30$ | | |
|---|---|---|---|---|---|---|---|
| $i$ | 1 | 2 | 3 | | 1 | 2 | 3 |
| $p_i$ | 0.3250 | 0.5500 | 0.7750 | $p_i$ | 0.2750 | 0.5167 | 0.7583 |
| $CR_i$ | 0.7011 | 0.8571 | 0.9418 | $CR_i$ | 0.8815 | 0.9488 | 0.9801 |

Two variants based on Brest approach are labeled *Brestbin* and *Brestexp*. *Brestbin* uses binomial crossover with control parameters values recommended in [12], i.e. $\tau_1 = \tau_2 = 0.1$ and $F_l = 0.1$, $F_u = 0.9$. In the *Brestexp* variant, values of *CR* are distributed uniformly in $[CR_1,\ CR_3]$, see Table I. The random localization was not applied in these variants.

The last two variants of DE in tests combine both the competitive and Brest self-adaptive mechanisms. Two types of crossover compete according the rules described above, the parameters of crossover are the same as in *Brestbin* and *Brestexp*. Two pairs of *CR* and $F$ values are kept with each individual in population, the first pair is used for exponential crossover and the second pair for binomial one. Variant labeled *Brestcmp* is without random localization. In variant labeled *Brestcmprl* random localization is applied to the mutation.

Thus, four competitive DE variants, labeled *Cbin9rl*, *Cexp9rl*, *Cbin9exp9rl*, and *Cbin6exp6rl* are included into experimental comparison with four other variants labeled *Brestbin*, *Brestexp*, *Brestcmp*, and *Brestcmprl*.

## IV. Experiments and Results

### A. Standard Benchmark

*1) Test Functions:* Five commonly used functions [9], [3], [2] were applied as standard benchmark:

- Ackley function—multimodal, separable,

$$f(\boldsymbol{x}) = -20\exp\left(-0.02\ \sqrt{\tfrac{1}{d}\textstyle\sum_{j=1}^{d} x_j^2}\right) - \\ -\exp\left(\tfrac{1}{d}\textstyle\sum_{j=1}^{d}\cos 2\pi x_j\right) + 20 + \exp(1)$$

$x_j \in [-30, 30]$, $\boldsymbol{x}^* = (0, 0, \dots, 0)$, $f(\boldsymbol{x}^*) = 0$.

- Griewank function—multimodal, nonseparable,

$$f(\boldsymbol{x}) = \sum_{j=1}^{d} \frac{{x_j}^2}{4000} - \prod_{j=1}^{d} \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1$$

$x_j \in [-400, 400]$, $\boldsymbol{x}^* = (0, 0, \dots, 0)$, $f(\boldsymbol{x}^*) = 0$.

- Rastrigin function—multimodal, separable,

$$f(\boldsymbol{x}) = 10\,d + \sum_{j=1}^{d} [{x_j}^2 - 10\cos(2\pi x_j)]$$

$x_j \in [-5.12, 5.12]$, $\boldsymbol{x}^* = (0, 0, \dots, 0)$, $f(\boldsymbol{x}^*) = 0$

- Rosenbrock function (banana valley)—unimodal, nonseparable,

$$f(\boldsymbol{x}) = \sum_{j=1}^{d-1} \left[100(x_j^2 - x_{j+1})^2 + (1 - x_j)^2\right]$$

$x_j \in [-2.048, 2.048]$, $\boldsymbol{x}^* = (1, 1, \dots, 1)$, $f(\boldsymbol{x}^*) = 0$.

- Schwefel function—multimodal, separable, the global minimum is distant from the next best local minima,

$$f(\boldsymbol{x}) = -\sum_{j=1}^{d} x_j \sin(\sqrt{\mid x_j \mid})$$

$x_j \in [-500, 500]$, $\boldsymbol{x}^* = (s, s, \dots, s)$, $s = 420.968746$, $f(\boldsymbol{x}^*) = -418.982887\,d$

Two levels of problem dimension were chosen, $d = 10$ and $d = 30$. The names of test functions (or their self-explaining abbreviations) are used as labels when reporting the results.

*2) Experiments and Results:* One hundred of independent runs were carried out for each function and level of $d$. The search was stopped if the difference between maximum and minimum function values was less than $1e{-}6$ or number of function evaluations (hereinafter *ne*) exceeded the limit $20000\,d$. Population size was set to 40 for problems with $d = 10$ and 60 for for problems with $d = 30$. Parameters controlling the competition of settings were set to $n_0 = 2$, and $\delta = 1/(5\,H)$.

Number of function evaluations *ne* and the success of search were recorded in each run. The search was considered successful, if $f_{min} - f(\boldsymbol{x}^*) < 1e - 4$, where $f_{min}$ is the minimum value of objective function in final population. The reliability (estimator of convergence probability), hereinafter $R$, was evaluated as the percentage of successful runs.

The performance of variants were compared using Q-measure. The Q-measure was proposed by Feoktistov [7] to integrate time demand and reliability into a single criterion of convergence. The formula of Q-measure is

$$Q_m = \overline{ne}/R, \tag{6}$$

where $\overline{ne}$ is the average of number of function evaluations in successful runs and $R$ is the reliability in %.

TABLE II
COMPARISON OF VARIANTS USING THE VALUES OF Q-MEASURE.

| $d$ | Variant | Ackley | Griewank | Rastrig | Rosen | Schwefel | Average |
|---|---|---|---|---|---|---|---|
| 10 | Cbin9rl | 176 | 179 | **152** | **250** | 135 | 178 |
| 10 | Cexp9rl | 188 | 183 | 198 | 286 | 142 | 199 |
| 10 | Cbin9exp9rl | 181 | 183 | 165 | **245** | 140 | 183 |
| 10 | Cbin6exp6rl | 157 | **156** | **150** | 272 | **122** | **172** |
| 10 | Brestbin | 152 | 161 | 198 | 717 | 125 | 270 |
| 10 | Brestexp | 152 | 169 | 192 | 680 | 126 | 264 |
| 10 | Brestcmp | **148** | 157 | 191 | 852 | 123 | 294 |
| 10 | Brestcmprl | **108** | **152** | 157 | 308 | **96** | **164** |
| 30 | Cbin9rl | 911 | 688 | **793** | 2542 | 746 | 1136 |
| 30 | Cexp9rl | 822 | 626 | 1504 | **1511** | 848 | 1062 |
| 30 | Cbin9exp9rl | 898 | 678 | 846 | 1756 | 758 | 987 |
| 30 | Cbin6exp6rl | 763 | 574 | **781** | 1601 | 661 | **876** |
| 30 | Brestbin | 583 | 477 | 1596 | 3777 | 607 | 1408 |
| 30 | Brestexp | 574 | 525 | 1393 | 1953 | 632 | 1015 |
| 30 | Brestcmp | **557** | **453** | 1431 | 2226 | **587** | 1051 |
| 30 | Brestcmprl | **434** | **420** | 1131 | **1455** | **494** | **787** |

Best and second best values of $Q_M$ are bold, the best values are underlined.

TABLE III
RELIABILITY OF THE SEARCH—VALUES OF $R$.

| $d$ | Variant | Ackley | Griewank | Rastrig | Rosen | Schwefel | Average |
|---|---|---|---|---|---|---|---|
| 10 | Cbin9rl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 10 | Cexp9Rrl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 10 | Cbin9exp9rl | 100 | 99 | 100 | 100 | 99 | 99.6 |
| 10 | Cbin6exp6rl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 10 | Brestbin | 100 | 96 | 99 | 98 | 99 | 98.4 |
| 10 | Brestexp | 100 | 89 | 100 | 100 | 100 | 97.8 |
| 10 | Brestcmp | 100 | 96 | 100 | 100 | 99 | 99.0 |
| 10 | Brestcmprl | 99 | 64 | 95 | 96 | 93 | 89.4 |
| 30 | Cbin9rl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 30 | Cexp9rl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 30 | Cbin9exp9rl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 30 | Cbin6exp6rl | 100 | 100 | 100 | 100 | 100 | 100.0 |
| 30 | Brestbin | 100 | 97 | 100 | 100 | 98 | 99.0 |
| 30 | Brestexp | 100 | 82 | 100 | 98 | 98 | 95.6 |
| 30 | Brestcmp | 100 | 95 | 100 | 100 | 100 | 99.0 |
| 30 | Brestcmprl | 85 | 66 | 100 | 93 | 82 | 85.2 |

The values of Q-measure for all variants and test functions are shown in Table II. The best and second best values of $Q_m$ are printed bold, the best values are underlined. Comparing the performance of self-adaptive variants of DE, *Brestcmprl* was the best in 7 of 10 test problems and its averages of $Q_m$ are the smallest for both dimension levels, but it does not appeared within best two variants in some problems (Rosenbrock, problem dimension $d = 10$, Rastrigin, both levels of dimension).

Reliability of the search is shown in Table III. Comparing the values of $Q_m$ in Table II and reliability in Table III, it is apparent, that higher efficiency of *Brestcmprl* is paid by less reliability of the search by this algorithm. Overall view shows that competitive variants are more reliable, but sometimes less efficient than the variants based on the evolutionary self-adaptive mechanism proposed by Brest et al.

### *B. Composition Functions*

The novel hybrid benchmark functions were proposed by Liang et al. [19], in order to make testing more similar to real-world problems. Commonly used standard benchmark problems are criticized due to the fact that their global minimum points have the same coordinate values in each dimension (the case of all functions in our standard benchmark), or the global minimum point is in the domain center of gravity (the case of Ackley, Griewank, and Rastrigin). The novel hybrid benchmark functions are not featured with any kind of such symmetry. Each of them is composed from several various functions by non-trivial transformations including matrix rotation. Thus, they are non-separable functions with many local minima, and the global minimum point cannot be found easily.

Six evolutionary algorithms were tested in this composite functions benchmark in [19]. Among these algorithms, comprehensive learning particle swarm optimization (CLPSO) was the most efficient in CF1, CF2, CF3, and CF5 problems, standard PSO algorithm in CF4 problem, and a variant of DE with exponential crossover in CF6 problem. Details and references are given in Liang et al.

*1) Experiments and Results:* The same eight variants of self-adaptive DE tested with standard benchmark were applied to hybrid benchmark functions. To enable the comparison of our results with the experimental results of the algorithms in [19], the arrangement of experiments was the same, i.e. for each test function, each algorithm was run 20 times and maximum function evaluations are set to 50,000 for all the algorithms. The domains are $D = \prod_{j=1}^{10}[-5, 5]$ for all test functions. Population size was set up to $N = 50$ for all the algorithms in tests. Other control parameters had values used for standard benchmark.

The averages and standard deviations of computed minimum function values of 20 runs are reported in Table IV respectively. Results of the best algorithm from [19] are reported, as well. The values less than the best ones found in [19] are printed bold, the least value for each function is underlined. The value 0 in CF1 column stands for the positive values less than 4.94e-324, which is the accuracy of double-precision arithmetic used in the implementation environment.

From eight new self-adaptive variants of DE, at least four variants outperformed the most successful algorithm in [19] for each function except CF6. In the case of CF4 even all seven variants were more efficient than the best algorithm reported in [19]. For CF1 problem, five self-adaptive variants achieved better results than those reported former, but the solution found by CLPSO in [19] is acceptable, so the improvement attained by self-adaptive variants is not important. Surprisingly, the bold values of standard deviation in Table IV are scarce, what implies that only small part of self-adaptive DE variants is more robust than the best algorithms from [19].

There is no unique winner among the DE variants tested on composition benchmark problems. However, we can con-

TABLE IV
RESULTS ACHIEVED BY THE ALGORITHMS ON SIX COMPOSITION FUNCTIONS

Averages of computed minimum function values in 20 runs

| Algorithm | CF1 | CF2 | CF3 | CF4 | CF5 | CF6 |
|---|---|---|---|---|---|---|
| Best in [19] | 5.7e-08 | 19.2 | 133 | 314 | 5.37 | 491 |
| Cbin9rl | **0** | **5.7** | **103** | **285** | **5.10** | 560 |
| Cexp9rl | **0** | 21.6 | **108** | **305** | **0.06** | 585 |
| Cbin9exp9rl | **0** | 21.0 | **100** | **285** | 10.27 | 580 |
| Cbin6exp6rl | 10 | **15.4** | **111** | **296** | **5.13** | 656 |
| Brestbin | 5 | **10.3** | 133 | **274** | **0.06** | 535 |
| Brestexp | **0** | **10.8** | **104** | **270** | 10.47 | 540 |
| Brestcmp | 5 | **6.1** | **106** | **271** | 10.24 | 500 |
| Brestcmprl | 20 | 31.9 | **112** | **285** | 15.68 | 596 |

Standard deviations of computed minimum function values in 20 runs

| Algorithm | CF1 | CF2 | CF3 | CF4 | CF5 | CF6 |
|---|---|---|---|---|---|---|
| Best in [19] | 1.0e-07 | 14.7 | 20.0 | 20.1 | 2.61 | 39.5 |
| Cbin9rl | **0** | 22.2 | 32.8 | **13.7** | 22.34 | 147.4 |
| Cexp9rl | **0** | 51.7 | 47.6 | 71.5 | **0.25** | 149.5 |
| Cbin9exp9rl | **0** | 51.9 | 35.4 | **9.3** | 30.69 | 165.1 |
| Cbin6exp6rl | 30.8 | 36.5 | 39.6 | 72.4 | 22.33 | 207.6 |
| Brestbin | 22.4 | 30.7 | 35.9 | **13.3** | **0.29** | 127.5 |
| Brestexp | **0** | 30.5 | 24.5 | 23.6 | 30.63 | 123.7 |
| Brestcmp | 22.4 | 22.1 | 41.5 | **14.8** | 30.70 | **0.1** |
| Brestcmprl | 52.3 | 56.4 | 28.5 | **11.1** | 36.35 | 183.0 |

clude that the novel self-adaptive DE variants were able to outperform advanced evolutionary algorithms in the majority of these hard test problems.

## V. CONCLUSIONS

Both self-adaptive patterns used in tested variants have proved the efficiency. All eight novel self-adaptive DE variants outperformed standard evolutionary algorithms in some test problems.

These results were achieved with implicit setting of control parameters without tuning their values. For given subset of benchmark problems the population size was set the same for all the DE variant. The influence of population size on algorithm efficiency was not examined in this study. This question should be solved in future including an attempt to solve some self-adaptive mechanism of population size.

The use of stand-alone exponential crossover increases efficiency in comparison with binomial crossover only for small part of problems. Applying competitive parameter setting together with both crossover types brings better convergence in standard benchmark problems. The combination of both self-adapting approaches results in the most efficient algorithm for standard benchmark problems. In the case of composition benchmark, the combination of both crossover types was beneficial for less part of problems.

Summarizing the results from both benchmarks, there is no generally winning variant. This is an implication resulting from the No Free Lunch Theorems [20], stating that any stochastic search algorithm cannot outperform others, when all possible objective functions are taken into account. Nevertheless, it does not exclude the possibility to propose novel variants of self-adaptive algorithms with better efficiency for wider range of objective functions and the self-adaptive variants of DE seem to be good examples of such algorithms advisable to the application in real-world problems of global optimization.

## ACKNOWLEDGMENT

Supported by the grant 201/08/0472 of the Czech Grant Agency and by the research project MSM 6198898701.

## REFERENCES

[1] A. P. Engelbrecht, *Computatinal Intelligence: An Introduction*. Chichester: John Wiley & sons, 2007.

[2] R. Storn and K. V. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, pp. 341–359, 1997.

[3] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

[4] P. Kaelo and M. M. Ali, "A numerical study of some modified differential evolution algorithms," *European J. Operational Research*, vol. 169, pp. 1176–1184, 2006.

[5] D. Zaharie, "A comparative analysis of crossover variants in differential evolution," in *Proceedings of IMCSIT 2007*, U. Markowska-Kaczmar and H. Kwasnicka, Eds. Wisla: PTI, 2007, pp. 171–181.

[6] ——, "Critical values for the control parameter of the differential evolution algorithms," in *MENDEL 2002, 8th International Conference on Soft Computing*, R. Matoušek and P. Ošmera, Eds. Brno: University of Technology, 2002, pp. 62–67.

[7] V. Feoktistov, *Differential Evolution in Search of Sotution*. Springer, 2006.

[8] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," in *Advances in Intelligent Systems Fuzzy Systems, Evolutionary Computing*, A. Grmela and N. E. Mastorakis, Eds. Athens: WSEAS Press, 2002, pp. 293–298.

[9] M. M. Ali and A. Törn, "Population set based global optimization algorithms: Some modifications and numerical studies," *Computers and Operations Research*, vol. 31, pp. 1703–1725, 2004.

[10] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algortithm," *Soft Computing*, vol. 9, pp. 448–462, 2005.

[11] A. K. Quin and P. N. Suganthan, "Self-adaptive differential evolution for numerical optimization," in *IEEE Congress on Evolutionary Computation*, 2005, pp. 1785–1791.

[12] J. Brest, S. Greiner, B. Boškovič, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 646–657, 2006.

[13] J. Tvrdík, "Competitive differential evolution," in *MENDEL 2006, 12th International Conference on Soft Computing*, R. Matoušek and P. Ošmera, Eds. Brno: University of Technology, 2006, pp. 7–12.

[14] ——, "Adaptation in differential evolution: A numerical comparison," *Applied Soft Computing*, 2007, submitted.

[15] ——, "Differential evolution with competitive setting of its control parameters," *TASK Quarterly*, vol. 11, pp. 169–179, 2007.

[16] J. Tvrdík, V. Pavliska, and H. Habiballa, "Stochastic algorithms for global optimization—matlab and c++ library," University of Ostrava, 2007. [Online]. Available: http://albert.osu.cz/oukip/optimization/

[17] V. Kvasnička, J. Pospíchal, and P. Tiňo, *Evolutionary Algorithms*. Bratislava: Slovak Technical University, 2000, (In Slovak).

[18] J. Tvrdík, "Exponential crossover in competitive differential evolution," in *MENDEL 2008, 14th International Conference on Soft Computing*, R. Matoušek, Ed. Brno: University of Technology, 2008, pp. 44–49.

[19] J. J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Proc. IEEE Swarm Intelligence Symposium*, 2005, pp. 68–75, matlab codes on web. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan/index_files/comp-functions.htm

[20] D. H. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67–82, 1997.

Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 933–937

ISBN 978-83-60810-14-9
ISSN 1896-7094

# Solving IK problems for open chains using optimization methods

Krzysztof Zmorzynski
Warsaw University of Technology
Faculty of Mathematics and Information Science
Email: zmorzynskik@student.mini.pw.edu.pl

***Abstract*—Algorithms solving inverse problems for simple (open) kinematic chains are presented in this paper. Due to very high kinematics chains construction complexity, those algorithms should be as universal as possible. That is why optimization methods are used. They allow fast and accurate calculations for arbitrary kinematics chains constructions, permitting them to be used with success in robotics and similar domains.**

## I. Introduction

THE problem of finding kinematic chain configuration for given position of its effector is called inverse kinematics (IK) problem. One of the major application of solving IK problem is robotics: having destination translation and orientation of the end effector, robot joints configuration should be found to achieve this translation and orientation. Robot mechanical constraints (for arms and joints) should be taken into consideration.

For some special cases, analytical and geometric methods for solving IK problems effectively exists [7], [9]. General explicit solution does not always have to exist, however. Also, as presented in [4], obtaining analytical solution might be very laborious and complex. These are the main drawbacks of analytical methods.

Methods based on Jacobian matrix and its inverse are widely described in literature [6], [8], [10]. When Jacobian matrix is not invertible, transposition or pseudoinverse Jacobian matrix is used instead [6], [8]. Computational cost of Jacobian matrix operations might be very high, though, not to mention stability problems in the neighborhoods of singularities.

In this paper optimization methods to solve IK problem are used. In contrast to methods mentioned above, they allow full automatization and perform all computations on a machine. They are also very universal, so can be used to solve arbitrary IK problem for open chains. Badler [1] used these methods to solve IK problem for more complicated chains containing many effectors, such as human skeleton. However, computation cost turned out to be to high to be used in real time, or even close to real time. Kim, Jang, Nam [2] used optimization methods to solve IK problem for binary manipulators (that is, each joint can take one of two possible positions).

More general manipulators are considered in this paper, with joints able to take any value from specified bounds. Supplied application allows user to modify chain construction, set effector's target translation and orientation and a real time visualisation of chain state during computations. It is a robust method to validate chain construction and check if desired target position can be achieved.

## II. Chain representation

### A. Chain configuration space

Open chains with each joint having single degree of freedom are considered in this paper. Whole chain configuration can be written as:

$$c = (q_1, q_2, \ldots, q_n) \in C \tag{1}$$

where $C$ is chain configuration space, $q_i$ is $i-th$ joint parameter and $n$ is number of joints.

### B. Joints

Two types of joints are specified: prismatic joints and revolute joints. Parameter value of each joint has to be between minimum and maximum values $q_i^{min}$ and $q_i^{max}$. Joints are connected to each other along their local $Z$ axes. It is also possible to set constant offset value along this axis (as presented in fig. 1). This is slightly simpler representation than in [5].

In the case of **prismatic** joint, parameter is additional (beside fixed offset mentioned above) length along local joint $Z$ axis.

In the case of **revolute** joint, parameter is the rotation angle (in degrees) along specified local rotation axis. It is clear, that in order to model revolute joint with greater DOF, it is sufficient to create many revolute joints, with their fixed offsets set to $0$, each allowing rotation about proper axis.

Beside parameter, each joint is represented also by its position, vector $g = (v, r)$, where $g$ is element from space $L = \Re^3 \times S^3$. $v$ is translation in world coordinates, $r$ is rotation quaternion also in world coordinates. Joints translations and orientations are calculated in forward kinematics manner, starting from first (or root) joint. Formally, position of the effector is the map:

$$\begin{aligned} E : C &\rightarrow L \\ c \in C &\rightarrow E(c) \in L \end{aligned} \tag{2}$$

The use of quaternions instead of rotation matrices allows faster computations.

Effector's target translation and orientation is also represented by element $e$ form space $L$.

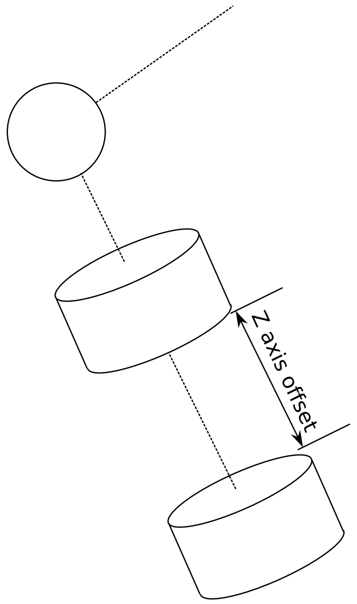


Fig. 1. Sample configuration. Dotted lines represent offsets along local Z axes.

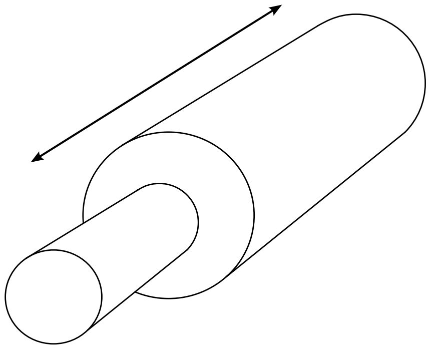


Fig. 2. Prismatic joint

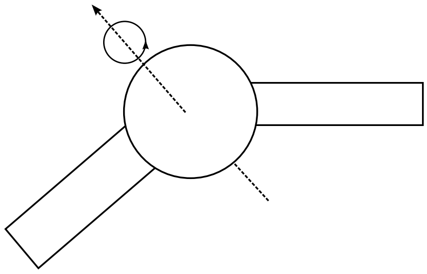


Fig. 3. Revolute joint. Dotted line is a rotation axis.

## III. Optimization methods

### A. Objective functions

In the case of solving IK problem, objective functions are generally distance (potential) functions from effector's current translation and orientation to effector's target translation and orientation. Written in a formal way, it is a function:

$$\begin{aligned} d : L \times L &\rightarrow \Re^+ \\ (x, y) \in L \times L &\rightarrow d(x, y) \in \Re^+ \end{aligned} \tag{3}$$

where $x$ and $y$ are two elements from space $L$.

Following objective functions can be distinguished:

- translation only difference - Euclidean metric can be used:

  $$d_{pos}(x, y) = \sqrt{(x_v - y_v) \cdot (x_v - y_v)} \tag{4}$$

  where $\cdot$ is dot product in $\Re^3$. Square root can be omitted in calculations.
- orientation only difference:

  $$d_{rot}(x, y) = \sqrt{(x_r - y_r) \cdot (x_r - y_r)} \tag{5}$$

  where $\cdot$ is dot product of unit quaternions in $S^3$.
- translation and orientation difference at the same time:

  $$d(x, y) = w_p d_{pos} + w_r d_{rot} \tag{6}$$

  where $w_p$ and $w_r$ are weights for translation and orientation functions, respectively.

### B. Optimization methods and calculations

To minimise objective function, one can use gradient based methods, such as **conjugate gradient method**. It converges faster than steepest descent method since it takes conjugate (with respect to gradient) vector when choosing minimisation direction. Fletcher-Reeves formula for computing ”direction factor” $\beta$ can be used (see [3]). When $\beta$ equals $1$ it means that conjugate direction is no longer valid and should be set to $0$ to make it steepest descent method.

Quasi-Newtonian methods, like **BFGS**, are even better than conjugate gradient. Although they require Hessian matrix computation, which may be expensive, it is also possible to use an approximation based on a function gradient value (described in [13]). The BFGS algorithm is presented in [12]. Its implementation available at [11] was used in the application.

One of the above functions, name it $d$ (4-6), is chosen to be minimised. It takes current and target position of the effector. Recall that from eq. 2, its position is based on current chain configuration:

$$\begin{aligned} c &\in C \\ e = E(c) &\in L \end{aligned} \tag{7}$$

where $c$ is chain configuration.

It could also be written, that the following function $f$ is being minimised:

$$\begin{aligned} f : C &\rightarrow \Re^+ \\ c \in C, y \in L &\rightarrow f(c) = d(E(c), y) \in \Re^+ \end{aligned} \tag{8}$$

where $c$ is current chain configuration, $y$ is the effector's target position. The function $f$ is minimised with respect to chain configuration and that means joints parameters. Joints parameters constraints should be taken into consideration while performing calculations (see next section).

### C. Gradient computation

Gradient of the function $f$ for conjugate gradient method is calculated using difference quotient (**two point**) method:

$$\bigtriangledown f(c) = (\frac{\partial f}{\partial q_1}(c), \ldots, \frac{\partial f}{\partial q_n}(c)) \tag{9}$$

where $c = (q_1, \ldots, q_n) \in C$ and $\frac{\partial f}{\partial q_i}$:

$$\frac{\partial f}{\partial q_i}(c) = \frac{f(q_1, \ldots, q_i + h, \ldots, q_n) - f(q_1, \ldots, q_i, \ldots, q_n)}{h}$$

where $h$ is appropriate small value (see IV on how this value could be chosen). While calculating partial derivatives $\frac{\partial f}{\partial q_i}$, joints constrains $q_i^{min}$ and $q_i^{max}$ are taken into account. It is ensured that:

$$q_i^{min} \leq q_i + h \leq q_i^{max} \tag{10}$$

When value $q_i + h$ is out of the bounds, it is truncated to $q_i^{min}$ or $q_i^{max}$.

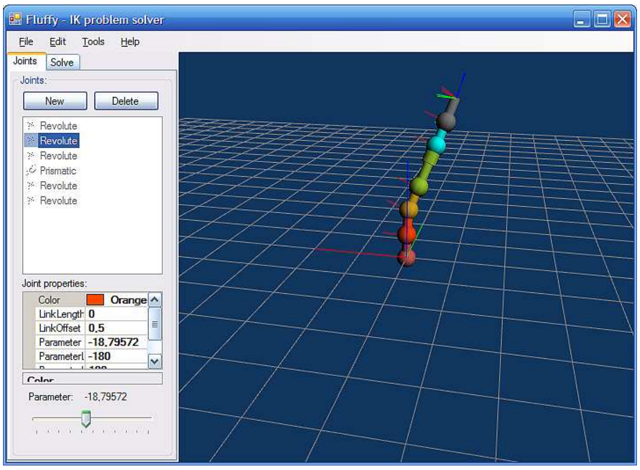
The use of BFGS method requires more precision in gradient computation. **Four point** method is used:

$$\begin{aligned} \frac{\partial f}{\partial q_i}(c) &= \frac{1}{12h}(f(q_1, \ldots, q_i - 2h, \ldots, q_n) \\ &- 8f(q_1, \ldots, q_i - h, \ldots, q_n) \\ &+ 8f(q_1, \ldots, q_i + h, \ldots, q_n) \\ &- f(q_1, \ldots, q_i + 2h, \ldots, q_n)) \end{aligned} \tag{11}$$

Thanks to definition of $f$ like in eq. 8 it is possible to move from the original IK problem to the problem of multivariable objective function minimisation.

## IV. Application

Created application allows user to interactively create and modify chain construction, save it to a file and load it later. Modification of joints parameters result in real time changes in chain visualisation. This applies to not only when user manipulates the parameters, but also while solving IK problem so it could be seen how the solution is being found. For given chain construction and end effector's target position, user chose which data types should be used in computations (see below), which objective function to minimise (see III-A) and which optimization method to use (see III-B). When calculations are finished, it is possible to view objective function value dependence on computation time and on iteration number.



Revolute
Revolute
Revolute
Prismatic
Revolute
Revolute

Joint properties:

| | |
|---|---|
| Color | 🟧 Orange |
| LinkLength | 0 |
| LinkOffset | 0,5 |
| Parameter | -18,79572 |
| ParameterL | -180 |

Color

Parameter: -18,79572

Fig. 4. Application screenshot.

The application is written in `C# 2.0` language. Calculations are based on `generics` objects, so every floating point number type in this language: 32 bit `float`, 64 bit `double` and 128 bit `decimal` can be used in computations. This allows to easily tell how much each data type has impact on the efficiency and precision of calculations.

When calculating function gradient as stated in III-C, it is crucial to chose right value for $h$. As it turns out, it is not always when the smallest value performs best. ”Batch” mode is available which allows user to compute with different values for $h$, so it could be estimated best for given chain construction and chosen data type.

By using the `.net 2.0` platform, it is possible to add new objective functions and optimisation methods solvers without need to recompile whole application. They are dynamically read at application startup from external `.net` dynamic link libraries (DLL) and added to internal database.

User can chose between `DirectX` or `OpenGL` graphics libraries used to render the scene. It is also possible to resign from rendering at all, increasing the same application performance.

## V. Examples

Following examples were computed on a AMD Turion 64 X2 (two cores, each 1.6GHz) notebook, on Windows XP 32 bit with 1GB RAM and .net 2.0. Optimization algorithm was running on a separate thread, which also means on the other core than main application thread. Null renderer was selected to avoid unnecessary rendering overhead.

All calculations starts with all joints parameters set to $0$ (fig. 5). Objective function 6 was used, with $w_p$ set to $0.5$ and $w_r$ to $1$.

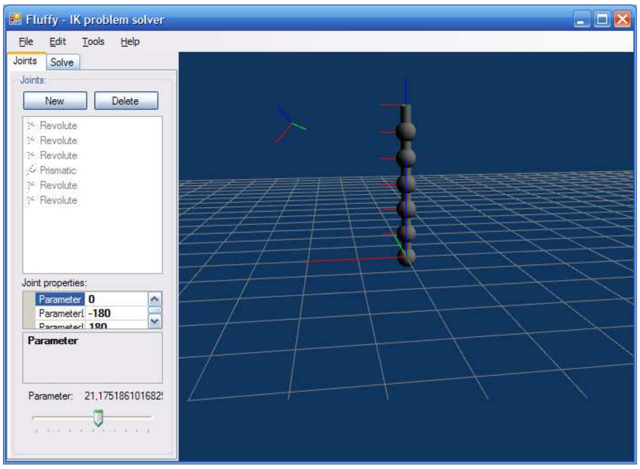
Today's widely used robots have mechanical arms precision set at about $10^{-4}$ and $10^{-5}$ [m]. IK problems solvers should be able to find chain configuration to achieve one of those precision. In this paper both values were researched.

### A. Example 1—relatively close to initial configuration

Chain with 6 joints was used, listed in order from the first (root) joint: first three revolute joints with rotation axes around $X$, $Y$, $Z$, respectively, then prismatic joint and then two additional revolute joints with rotation axes $X$ and $Z$. All revolute joints has parameters constraints set to $(-90, 90)$ degrees, where prismatic joint has $(0, 2)$.

Effector's target position was set relatively close to chain's initial configuration. Figure 6 presents final calculated configuration.

All three data types were investigated, with $h$ value (see III-C) set in batch mode. Tables I, II presents best times and values for each data type for $10^{-4}$ objective function precision,



| | |
|---|---|
| Parameter | 0 |
| Parameterl | -180 |
| Parameterl | 180 |

Parameter

Parameter: 21,175186101682

Fig. 5. Starting configuration.

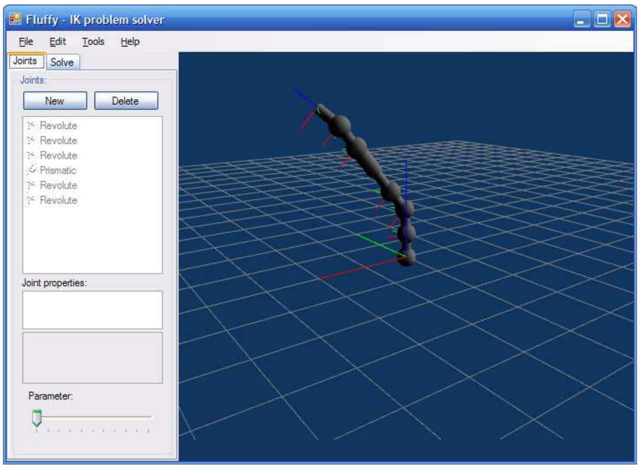


Fig. 6. Example 1 solution.

for conjugate gradient method and BFGS, respectively. Results for $10^{-5}$ are presented in tables III and IV.

TABLE I
EXAMPLE 1 RESULTS FOR $10^{-4}$ AND CONJUGATE GRADIENT METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-5}$ | 80 | 35 |
| `double` | $10^{-10}$ | 75 | 65 |
| `decimal` | $10^{-12}$ | 80 | 300 |

TABLE II
EXAMPLE 1 RESULTS FOR $10^{-4}$ AND BFGS METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-2}$ | 45 | 25 |
| `double` | $10^{-6}$ | 45 | 45 |
| `decimal` | $10^{-2}$ | 40 | 230 |

It can be seen from the tables I, II, III and IV, that `float` and `double` types performs best, due to their hardware support. `float` is fastest, probably because of smaller loading times from memory that `double`. Number of iterations for all types are similar. `decimal` type has significant calculation impact—nearly 10 times slower than `float`, mainly because of its software `.net` emulation.

TABLE III
EXAMPLE 1 RESULTS FOR $10^{-5}$ AND CONJUGATE GRADIENT METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-4}$ | 105 | 48 |
| `double` | $10^{-10}$ | 105 | 80 |
| `decimal` | $10^{-12}$ | 90 | 320 |

TABLE IV
EXAMPLE 1 RESULTS FOR $10^{-5}$ AND BFGS METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-2}$ | 50 | 30 |
| `double` | $10^{-10}$ | 40 | 45 |
| `decimal` | $10^{-2}$ | 45 | 250 |

Clearly, BFGS method is faster by about 50% than conjugate gradient method. It also needs half of the iterations to find the solution.

Figure 7 presents rough overview of how convergence was achieved in the matter of iterations (left plot) and time (right plot).

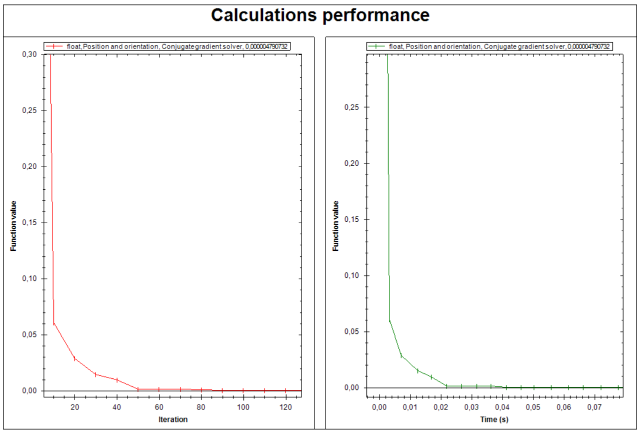


Fig. 7. Example 1 convergence overview.

## B. Example 2

This time effector's target position was farther than in previous example (as in fig. 8), so computation time was expected to be longer. Also, to ensure that effector reaches target position, additional revolute joint was added at the end of the chain, and each revolute joint has constraints set to $(-180, 180)$ degrees. All revolute joints, except the first and the last one, have fixed offset set to $0$.

Results are presented in tables V and VII.

TABLE V
EXAMPLE 2 RESULTS FOR $10^{-4}$ AND CONJUGATE GRADIENT METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-3}$ | 105 | 50 |
| `double` | $10^{-6}$ | 90 | 65 |
| `decimal` | $10^{-5}$ to $10^{-12}$ | ˜110 | ˜400 |

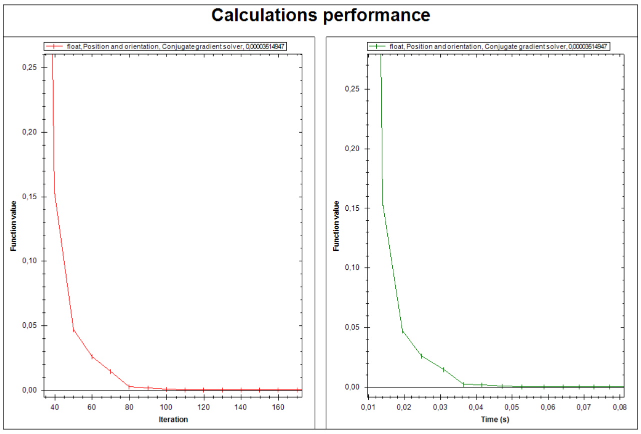


Fig. 9. Example 2 convergence overview.

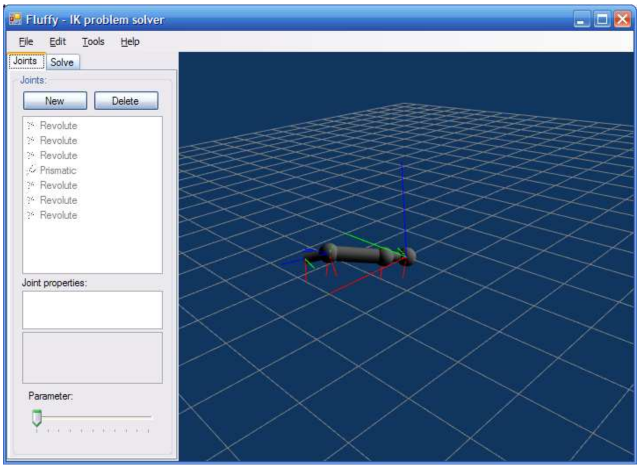


Fig. 8. Example 2 solution.

TABLE VI
EXAMPLE 2 RESULTS FOR $10^{-4}$ AND BFGS METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-2}$ | 40 | 25 |
| `double` | $10^{-1}$ | 35 | 45 |
| `decimal` | $10^{-1}$ to $10^{-12}$ | ˜35 – 40 | ˜250 |

TABLE VII
EXAMPLE 2 RESULTS FOR $10^{-5}$ AND CONJUGATE GRADIENT METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-3}$ | 120 | 58 |
| `double` | $10^{-6}$ | 100 | 76 |
| `decimal` | $10^{-5}$ to $10^{-12}$ | ˜120 | ˜500 |

TABLE VIII
EXAMPLE 2 RESULTS FOR $10^{-5}$ AND BFGS METHOD

| data type | $h$ | iterations | time (ms) |
|---|---|---|---|
| `float` | $10^{-3}$ | 45 | 30 |
| `double` | $10^{-1}$ | 40 | 50 |
| `decimal` | $10^{-1}$ to $10^{-12}$ | ˜35 – 40 | ˜300 |

This time, calculation times excludes `decimal` type from any practical usage. For conjugate gradient method, two other types performs slightly worst than in the previous example, but still in the almost real-time. For BFGS method, there are almost no differences between both examples. For all data types, iteration count is quite similar to each other.

Figure 9 shows how convergence was achieved.

## VI. CONCLUSIONS AND FUTURE WORK

It is clear from above examples, that optimization methods can be widely used to solve IK problems in the real time, even on commonly available customers computers. This leaves no need of use dedicated and expensive workstations, as it has been before.

Applying commonly used and available methods (such as conjugate gradient and BFGS) seems to work well in most cases. Their adaptation to multithreaded or multiprocessor environments may perform even better.

## REFERENCES

[1] J. Zhao and N. I. Badler, *Inverse kinematics positioning using nonlinear programming for highly articulated figures*, ACM Transactions on Graphics, Vol. 113, No. 4, October 1994, pages 313–336.

[2] Y. Y. Kim, G. W. Jang and S. J. Nam *Inverse kinematics of binary manipulators by the optimization method in continous variable space*, IEEE International Conference on Intelligent Robots and Systems, September 28 – October 2, 2004, Sendai, Japan.

[3] J. R. Shewchuk *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, School of Computer Science, Carnegie Mellon University, August 4, 1994.

[4] J. Angeles *Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms*, Springer, 2nd edition, 2003.

[5] L. T. Wang and B. Ravani *Recursive Computations of Kinematic and Dynamic Equations For Mechanical Manipulators*, IEEE Journal of Robotics and Automation, VOL. RA-1, NO. 3, September 1985.

[6] S. R. Buss, *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Dumped Least Squares methods*, University of California, San Diego, April 2004.

[7] Kang Teresa Ge *Solving Inverse Kinematics Constraint Problems for Highly Articulated Models*, Masters of Science thesis, University of Waterloo, 2000

[8] D. Park *Inverse Kinematics*, Computer Graphics, Department of Computer Science, University of Buenos Aires, Argentina.

[9] X. Wu et al. *A 12-DOF Analytic Inverse Kinematics Solver for Human Motion Control*, Journal of Information and Computational Science 1: 1 2004, pages 137–141.

[10] W. Stadler, P. Eberhard *Jacobian motion and its derivatives*, Mechatronics 11, 2001, pages 563–593.

[11] S. Bochkanov, V. Bystritsky *AlgLib - BFGS-B - http://www.alglib.net/optimization/lbfgsb.php*, 1999–2008.

[12] R. H. Byrd, P. Lu and J. Nocedal. *A Limited Memory Algorithm for Bound Constrained Optimization*, SIAM Journal on Scientific and Statistical Computing, 1995, 16, 5, pp. 1190–1208.

[13] J. Nocedal, S. J. Wright *Numerical Optimization*, Springer-Verlag, 1999.

# Author Index