

Assessment of Software System Presentation Layers Based on an ECORAM Reference Architecture Model

Michał Turek, Jan Werewka, Kamil Sztandera, Grzegorz Rogus
AGH University of Science and Technology
Al. Mickiewicza 30
Kraków, Poland
Email: {mitu, werewka, sztandera, rogus}@agh.edu.pl

Abstract– Software systems are constructed using many different solutions. In practice it is important to assess different software systems. The most effective method of comparing software systems is to use a predefined reference architecture model as the basis for comparison. In this paper an ECORAM presentation layer reference architecture model is proposed. ECORAM is an extended common reference architecture model which contains a crucial number of elements used in systems investigation. The reference architecture is described using ArchiMate notation. For each element of the reference architecture a search for a mapping to the selected solution architecture is performed. When mapping fails it is assumed that a given feature of the reference architecture doesn't exist in the target system. Comparisons can be based on different aspects and the quality of metrics for predefined aspects may be defined. In this paper a comparison examples of different visualization solutions of business intelligence and game systems are presented.

I. INTRODUCTION

SOFTWARE systems are characterized by different architecture solutions. The solution architecture should be based on a reference architecture which is a template for a family of software systems or specific domains of interest.

The reference architecture [1] models the abstract architectural elements in the domain of interest, independent of the technologies, protocols, and products that are used to implement a specific solution for the domain, and represents the essence of the architecture of a collection of systems [2]. The purpose of the reference architecture is to provide guidance on developing architectures for new versions of a system or extended families of systems and products.

The reference model [1] describes the important concepts and relationships in the domain, focuses on what distinguishes the elements of the domain, and the reference architecture elaborates further on the model to show a more complete picture.

One of the main perspectives on Reference Architecture Models is that they are knowledge repositories which facilitate knowledge transfer and communication. However, they can also serve as frameworks, lexicons of terms and naming conventions, as well as structural relation-

ships. One of the main challenges is to make the inherently abstract Reference Architecture Model concrete and understandable by providing sufficient specific information and guidelines.

When building software solution architecture in a given domain it is very important to have reference architecture. When obtaining software reference architecture it is important to have domain concepts and architecture reference models.

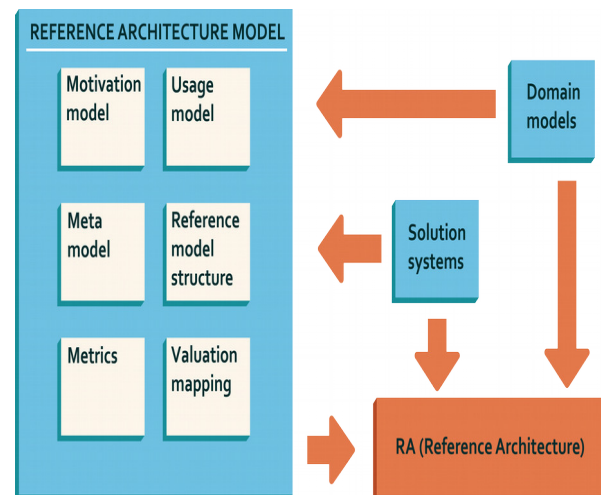


Fig 1. A reference architecture model in a reference architecture building context.

Domain models and solution systems may be used when building reference architecture models (fig. 1). The reference architecture model consists of the following elements:

- Motivation to build the reference architecture
- A usage model based on services (how the reference model will be used)
- A meta model defining the relationship in the set of related models.
- A reference model structure, consisting of elements and their descriptions.
- Metrics for defining the values of the reference model features.
- Valuation mapping. The elements of the reference architecture model can have different values for different stakeholders.

The reference architecture is developed based on a reference model, domain models and solution systems.

II. RELATED WORKS

The related works concerning the development of reference architecture models and the visualization capabilities of business intelligence systems are as follows.

Publication [3] describes a so-called RAModel (Reference Architecture Model) for reference architectures which can be included in reference architectures and presents all possible elements, broken down by type and relationship. The RAModel guides and improves productivity when new reference architectures are built and can offer effective support in the use and evolution of existing architectures. The RAModel also provides information on the elements and associated relationships which could be contained in reference architectures, independently from the application domains or purposes of such architectures. In the paper [4] a ProSARA process is presented for building reference architectures, focusing on how to design, represent, and evaluate such architectures. Publication [5] provides a framework for classifying software reference architectures. This framework is based on three dimensions: goals (why something is defined), context (where it will be used, who defines it, when it is defined), and design (what it describes, how specific and detailed this description is, how is it represented). Two purposes for using reference architecture are distinguished to standardize specific architectures (aimed at the interoperability of systems/components) and to enable the design of specific architectures (aimed at providing guidelines for the design of systems in the form of blueprints, patterns, etc.).

The question arises concerning to what extent reference architecture can be used for general application. In [6], CORA (COMmon Reference Architecture) is described for general use, and examples of its utilization are given, but the work proposes more of a reference model than a reference architecture.

In Business Intelligence Systems Presentation Layer solutions start to play an important role. In paper [7], which concerns big data analytics, two views of data presentation are described:

- Data visualization. The goal of data visualization is to communicate information clearly and effectively. In BI systems data volume is increasing, so it is important to use new visualization solutions.

- Analytics of data originating from different sources. The goal of web analytics is to automatically retrieve, extract, and evaluate information for knowledge discovery from web services and documents. Web content involves several data types such as text, symbolic data, metadata, hyperlinks and multimedia data including image, audio, and video. Multimedia analytics has to deal with multimedia data. Social media content contains text, multimedia, locations and comments.

Paper [8] indicates that using more interactive visual methods in BI would be beneficial for users. The paper contains confirmation of the following statements: Most visual methods currently applied in Business Intelligence are static or employ only very limited forms of interactiv-

ity; Increasing the interactivity of visual methods is desired by users. According to users, interactivity helps them gain information and knowledge in business data analysis.

In article [9] the visualization capability of BI systems in terms of developing effective visualization for addressing business problems is crucial to the success of BI. To amplify decision makers' perception and cognition of business data, BI systems often deploy a variety of visualization techniques. In the paper a Context Adaptive Visualization (CAV) framework is proposed which can guide the design and implementation of the visualization systems to be integrated into or used together with BI systems. The system requirements for flexible visualization development are based on visual solutions like creation, modification enhancement, integration and transformation.

III. ECORAM PRESENTATION LAYER

Presentation plays a crucial role in IT systems. Historically, the presentation layer was separated from the application logic in order to enable different presentations without redesigning the whole application [10]. The presentation layer, especially the Graphical User Interface (GUI), has become a very important part of software for both users and developers [11]. There is a set of specific applications that relies mostly on User Interface (UI) and Human-computer interaction (HCI), with games and game controllers being among the most important representatives of this kind of application. From a design perspective, the presentation layer represents a specific set of problems that lead to a number of different solutions. The general idea of presentation layer design is to keep the presentation-related code separate from the domain-related code. The domain code should be completely unaware of the presentation logic in order to create reusable objects and components that are easy to test and maintain. The purpose of the presentation layer is to manipulate the presentation medium, give the user the look and feel experience (e.g. manipulate UI controls, present multimedia content), provide output to and receive input from the user.

The ECORAM (ECO Reference Architecture Model) is constructed from architecture building blocks (ABBs) that are comprised of information about the styles and architectural templates or features used [12]. The ECORAM is based on a layered model described in ArchiMate [13], so ABBs are grouped and presented in layers to emphasize their common properties. ArchiMate is an architecture description language (ADL), which is very suitable for communicating solutions among business and technology staff. The established approach assumes the developing of a new presentation model designed for product managers and system architects. This model should enhance understanding of user viewpoint dependencies and technical needs related to the entire service delivery in corporate architectures.

The main reason for developing the ECORAM was to express all system endpoint characteristics. Firstly, there

is a need to classify the device held by the user. The first branch of the ECORAM is extremely simplified, adjusted just to describing the general device type. Then, on a deeper level, a feature set must be considered. The user endpoint may be full-featured, reduced or extremely basic.

Once the presentation platform is ready, another matter should be considered, namely the general type of information processed. This information may be current or historic, concurrent or object-based, fully verified or possibly damaged, complete or only partial, and so on.

The final branch should be a broad classification of the interaction with the user. Here, all kinds of HCI, content exchange or presentation services will be considered.

As long as the system has a user interface, it can be classified using the ECORAM. Finally, after a long period of discussions and fixes, a basic model branch list was established. All model branches have been included in a compact set.

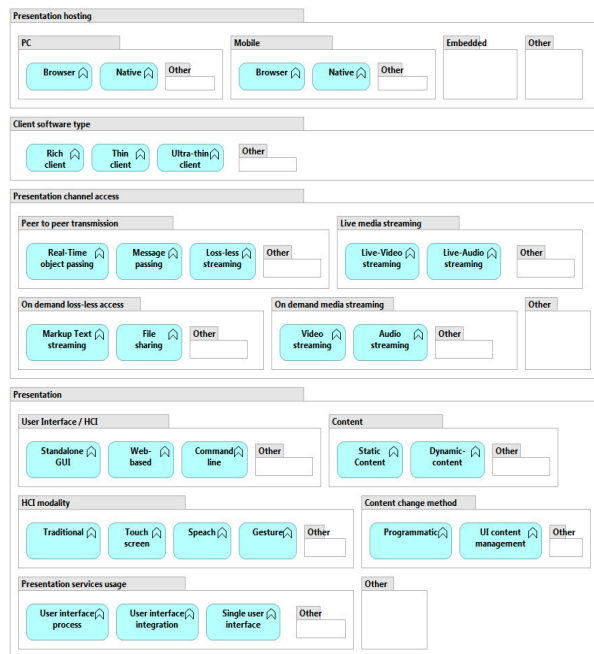


Fig 2. ECORAM presentation layer components in full set.

A. Presentation Hosting

The first selected group of entities concerns presentation hosting. It represents three types of physical devices used to display information:

- PC – Any kind of personal computer (desktop, notebook, notebook, etc.): Browser – an application running in a Web browser on a PC; Native Client – a native application running on a specific OS.

- Mobile – Any kind of mobile device working on a specific OS (mobile phone, smart phone, tablet, etc.): Browser – an application running in a Web browser on a mobile device; Native Client – an application running on a mobile device.

- Embedded – a system with specific hardware dedicated to performing a computing function within a mechanical or electrical system (Real Time Systems, con-

trollers, MP3 players, traffic lights, etc.). For embedded systems native, headless and other solutions can be used.

B. Client Software Type

This division specifies the set of features and capabilities available in the solution and the degree of its dependence on the operating system. Most importantly, client types can easily be divided into three kinds:

- Rich Client – a typical standalone application deployed on a user's machine (Workstations, i.e. desktop UI, etc.) or a RIA (Rich Internet Application) running in a Web Browser, usually with a graphical user interface displaying data using a range of controls. Suitable for disconnected and occasionally connected scenarios.

- Thin Client – an application that depends heavily on a server in order to provide functionality. All the computing is performed on the server side, which distinguishes this type of software from rich clients.

- Ultra-thin Client – Server Based Computing (Independent Computing Architecture), delivering only presentation data and/or sending user actions.

C. Presentation Channel Access

The next section specifies an abstract information flow between the user and the system. It should answer the question: "How do we deliver the presentation content?". Information can either be stored or retransmitted to a final user, or passed without any storage involvement. It can also be just a plain object, or possibly an infinite stream. Each user can also order just a part of a steam previously stored. Consequently, we propose the following list of presentation channel components:

- Peer to peer transmission - a transmission established for data generated in a real time. ECORAM assumes three kinds of such communication. 1) Real-Time object passing, occurring when an object is passed from the sender to the receiver in real time. These objects are based on patterns or classes and their size is limited. Objects can be cloned and sent to an unlimited number of receivers. (SOAP, REST etc.). 2) Message passing, messages can be read later (so storage is required). Messages may be cloned and delivered to a number of receivers without being deleted from storage (SMS, EMAIL, PAGER etc.). 3) Loss-less streaming, when a stream of data (taken in real time from a sensor or some such) is sent. It is encoded using dedicated protocol. The data size in this case is undefined.

- Live media streaming – a media stream does not require the use of any rigorous data quality assurance mechanisms. A stream may be defective or contain misrepresented data. In the case of this kind of stream, delay reduction is of the greatest importance: Live-Video streaming – Video content streaming; Live-Audio streaming – Audio content streaming.

- On demand lossless access – a stream with storage (on demand) – a limited size stream stored in data storage (similar to messages). Encoded using dedicated protocols: Markup Text streaming – a structured text described with mark-ups exchanged on demand between the sender and

the receiver (RSS, TELETEXT, etc.); File Sharing – Files exchanged on demand between the sender and the receiver.

- On Demand Media Streaming – a media stream with data storage – on receipt, no type of rigorous data quality assurance mechanisms is required. Video Streaming – Video content streaming; Audio streaming – Audio content streaming.

D. Presentation

A presentation mode, applied in a system interface, must deal with a user's (or human's) influence on that interface, and subsequently on the system itself. The following list divides interface presentation possibilities considering the type of user interaction:

- User Interface/HCI – The user interface is a part of a software program that allows the user to interact with computers and run their tasks. Human-computer interaction (HCI) describes the interaction between users and computers: Standalone GUI – a UI separated as a standalone application; Web-based – a User Interface based on a web application and characterized by request response navigation (interaction with the user); Command Line – Console-based applications offering an alternative text-only user experience, and typically run within command shells such as a Command window or a Power Shell. Most suitable for administrative or development tasks.

- HCI modality – the HCI modality is a general class of a sense through which the human can receive output from or give input to the computer: Traditional – Keyboard, mouse or touchpad; Touch Screen – Touch (multi-touch) or gestures on a Touch Screen; Speech – AUI (Audio User Interface) equipped with speech recognition solutions; Gesture – An interaction based on head and body movements

- Content – Describes whether the presented content will be changed by the user (dynamic content) or not (static content): Static content – content doesn't change, e.g. a static web-page; Dynamic Content – Presented content generated dynamically.

- Content change method – Describes how the presented content will be changed by the user: Programmatic – content refreshed or generated automatically, independently from the user's actions; UI Content management – Refreshed or generated depending on a user's actions, including Content Management Systems (CMS). A CMS is a computer program that allows content to be published, edited, modified and maintained from a central interface, and provides procedures to manage the workflow in a collaborative environment

- Presentation Services Usage – Describes the usage of different presentation services and user interactions between them: User Interface process – Synchronizes and orchestrates user interactions between presentation services for one actor and a one-time execution span [6]; User interface integration – Integrates different user interfaces and existing presentation services by offering various presentation integration services (i.e. WSRP, WS ren-

dering); The User Interface Integration itself behaves like a User Interface, providing a presentation service by integrating a 'real' presentation service [6]; Single User Interface – One user interface – no need for integration

As we can see, the ECORAM is optimized to just four main sections, enabling the quick classification of a system. On the other hand, internal section granularity will support the expressing of all vital differences between two or more presentation layer structures, even those used in completely different kinds of IT systems.

IV. ECORAM PRESENTATION LAYER APPLICATION IN BUSINESS INTELLIGENCE

ECORAM may be used for analyzing and developing human interactive computer systems (HICS). It is assumed that in BI systems interaction is performed in a timely manner. This means the reaction time is in accordance with human reaction. The application of an ECORAM for BI systems can be explained briefly as follows:

- The motivation is to use the reference model for a broad spectrum of applications. In particular, the model should be useful for BI systems.

- The model can be used in constructing BI systems, and also for investigating multimedia systems as potential data sources.

- The metamodel can be used as a default template for an application layer framework in an IT system of any kind. It shows meta-relations and meta-entities (here sections) that can be used or affected.

- Reference model structure, which is a composition of reference model elements. The description of model elements is also included here. The proposed structure can be applied for BI systems.

- Metrics defining the importance of the model features for BI systems.

- Valuation mapping – feature values important for BI System stakeholders.

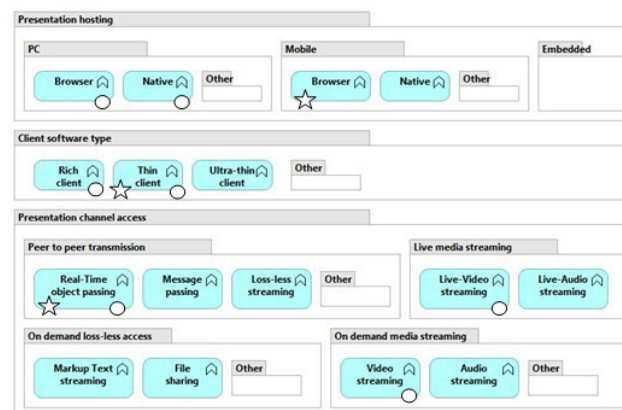


Fig 3. Sample mapping of asset usage: Taxi service management BI system (star symbols) and Integrated DVR Monitoring BI system (circle symbols).

Assessing software solution architecture is not an easy task. For the presentation layer some quality metrics can be distinguished, e.g. usability, user experience, and playfulness. ISO Standards 9126-3 [14] describe usability as product quality. According to ISO 9241-210 [15], "user

experience represents a person's perceptions and responses resulting from the use or the anticipated use of a product, system or service". In terms of user experience, playfulness can come from a design that engages people's attention or involves them in an activity for play, amusement, or creative enjoyment.

Fig. 3 shows the features used by two different BI systems and displays only a part of the ECORAM diagram. A full ECORAM diagram can be used to compare even totally different types of system, including different system domains. Each component in a solution's architecture is checked for features (assets) which also exist in the ECORAM. Various icon symbols are used to distinguish particular solution architectures (see Fig. 3). The diagram is used for a comparison of different systems. In a taxi business intelligence system [16] the customer uses a mobile device to connect to online taxi booking services and track the route of the approaching taxi. Additionally, by collecting information for many BI systems we can obtain statistical data which will give insight into how systems in a given domain are built. In [17] a Laboratory Management System (LIMS) delivering Real-Time Manufacturing Intelligence monitors a wide range of variables of both the manufacturing and the laboratory databases

IV. ECORAM PRESENTATION LAYER VIDEO GAMES APPLICATION

ECORAM may be used for game systems comparison. Finally, three different popular game systems were considered: Angry Birds, the World of Tanks, and Kinect Sports.

Angry Birds is a puzzle video game developed by the Finnish computer game development company Rovio Entertainment. Inspired primarily by a sketch of stylized wingless birds, the game was first released for iOS in December 2009. Since then, over 12 million copies of the game have been purchased from the iOS App Store, which has prompted the company to design versions for other touchscreen-based smartphones, most notably those using the Android, Symbian, Windows Phone and BlackBerry 10 operating systems. It has since expanded to video game consoles and for PCs.

The World of Tanks is a multiplayer online game developed by a Belarusian company called Wargaming.net, featuring mid-20th century fighting vehicles. The focus is on player vs. player game play with each player controlling a tank or an armored vehicle. The World of Tanks debuted as an eSports game at the World Cyber Games 2012. On 10 June 2013, it was announced that the World of Tanks was coming to the Xbox 360 in summer 2013 as the World of Tanks Xbox 360 Edition. Xbox 360 players will use servers separate from those playing the Windows version and players of each version will have separate accounts.

Kinect Sports is a sports video game developed by Rare and published by Microsoft Game Studios for Xbox 360. The game is a collection of six sports simulations and eight mini-games, designed to demonstrate the motion-sensing capabilities of Kinect. The six sports in-

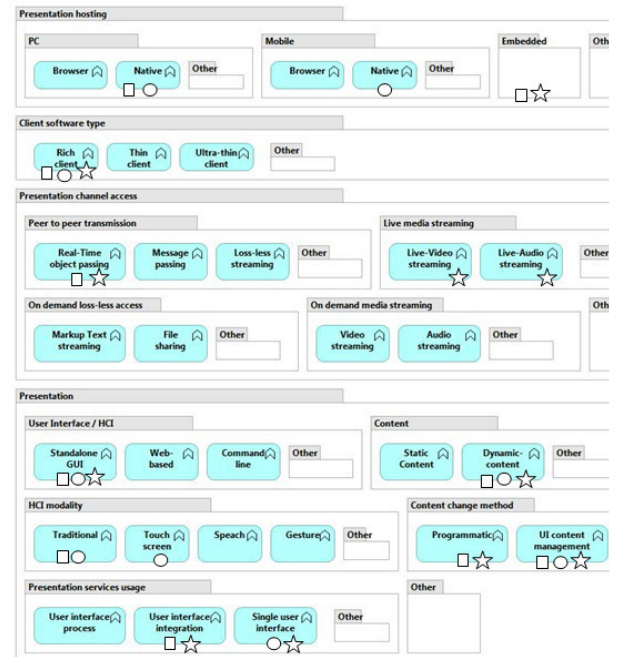


Fig 4. Sample system comparison: Kinect Sports (star symbols) and Angry Birds (circle symbols), World of Tanks (rectangle symbols).

cluded are: Bowling, Boxing, Track & Field, Table Tennis, Beach Volleyball and Football (Soccer in North America). Standing in front of a Kinect sensor, players compete by mimicking actions performed in real-life sports, such as throwing a javelin or kicking a football. The game received generally positive reviews from critics and sold over three million units as of April 2011.

The features used in the games can be evaluated from different viewpoints. The example considered here is playfulness. Playfulness motivates one to use the product and learn new features and technologies of the device. Paper [18] provides user interface characteristics influencing playfulness: creative enjoyment, challenge, curiosity, ability to customize the user interface, fun-in-doing, exploration, feedback, fantasy, metaphor and social interaction. A different perspective for understanding social game design is given by SoPlay heuristics [19]: accessibility – easy to approach, to understand and play; interrupt ability –spontaneous and irregular play sessions; continuity –a game world attracting the player to come back; discovery –new experiences and surprises; virility – growth in the player's social network; narrativity –elicits curiosity; expression –self-discovery and customization; sharing – collaborating with friends, sociability among friends, social competition with others.

We have selected the following key aspects for a playful user interface: Creative enjoyment, challenge, feedback; Accessibility – making the game easy to approach, understand and play; Ability to customize the user interface; Social interaction.

$$PF(\text{selected aspects}) = \sum_{i=1}^{CE \text{ aspects}} (\alpha_i CE_i) + \sum_{i=1}^{C \text{ aspects}} (\beta_i C_i) + \sum_{i=1}^{SI \text{ aspects}} (\gamma_i SI_i) \quad (1)$$

For the selected games, a weighted quality factor can be determined to assess the play ability of the presenta-

TABLE 1.
KEY ASPECTS OF A PLAYFUL (PF) USER INTERFACE

Key Aspect	Positive influence aspects (make use of)
Creative enjoyment (CE)	Touch screen, Speech, Gesture
Customization (C)	Dynamic content, UI content management
Social interaction (SI)	Live Video Streaming, Live Audio Streaming, User interface integration

tion layer solution. Assuming that all coefficients α_i , β_i , γ_i are equal to 1, the playfulness of each game stemming from the presentation layer (fig. 4) can be determined in conjunction with Table 2. The results are as follows: PF(Angry Birds aspects)=3, PF(World of Tanks aspects)=4, PF(Kinect Sports aspects)=6. The simplified calculation demonstrate that the Kinect Sports game is characterized by the highest playfulness. The calculation of quality factors demonstrate only an approach to comparing different systems based on a proposed model.

V. CONCLUSIONS

The process of ECORAM model development may be a part of knowledge management in a software development company [20]. The cooperative knowledge discovery model [21] including communication, coordination and cooperative decision making could be applied for developing and assessment of software solutions.

Reference architectures should accelerate work, reduce operating expenses and improve the quality of software system development, mainly due to reuse. Before reference architecture is designed it is important to develop an architecture reference model. To prove the flexibility of the model, a multiple system comparative analysis has to be performed, including system domains for auctions, games and BI. Each system chosen usually consists of many different presentation layer components that can be compared and placed in an ECORAM diagram. The diagram is capable of expressing all the differences with a sufficient level of detail.

Another possibility is to investigate the popularity of model elements in terms of usage by different user groups. For different kinds of users, we can now identify the presentation layer components used. The constant monitoring of a user group's activities and the system service counters will generate important statistical data for investment planning for future software line development. The solution will also make common feature exploration processes easier and will help system developers to design and refactor system architecture.

Finally, an ECORAM diagram can also be useful as a presentation tool during any kind of system requirements-establishment discussion with customers. The ECORAM diagrams are also used by computer science students when presenting their solution architectures, with the aim

of quickly identifying the class of a developed software system.

It is possible to provide a development time-line in ECORAM diagrams (by creating a diagram set with time-stamps) - and express a product line development regarding particular presentation layer components gradually involved. Building time-lined ECORAM diagrams for a set of similar products can easily express trends in product developments of that kind – helping to make a proper investment decisions in own product.

REFERENCES

- [1] Reference Architecture Foundation for Service Oriented Architecture Version 1.0, 2011, p 120, <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf>
- [2] Muller, G.: A Reference Architecture Primer. Buskerud University College, 2013, Gaudi documents, 2013, pp.21, www.gaudisite.nl
- [3] Nakagawa E. Y., Oquendo F., Becker M.: RAModel: A Reference Model for Reference Architectures, 2012 Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture
- [4] Nakagawa E. Y., Guessi M., Maldonado J. C., Feitosa D., Oquendo F.: Consolidating a Process for the Design, Representation, and Evaluation of Reference Architectures. Conference on Software Architecture (WICSA), 2014 IEEE/IFIP 2014, pp. 43 – 152
- [5] Angelov S., Grefen P., Greefhorst D.: A Classification of Software Reference Architectures: Analyzing Their Success and Effectiveness, 2009 IEEE/IFIP WICSA/ECSA
- [6] Elzinga T., van der Vlies J., Smiers L., The CORA Model, A Practical guide on using a COMmon Reference Architecture to design and deliver integrated IT solutions successfully, Sdu Customer Service, 2009
- [7] Han Hu , Yonggang Wen, Tat-Seng Chua, And Xuelong Li: Toward Scalable Systems for Big Data Analytics: A Technology Tutorial, IEEE Access, Vol. 2, 2014, pp. 652- 687
- [8] Aigner W.: Current Work Practice and Users' Perspectives on Visualization and Interactivity in Business Intelligence, 17th International Conference on Information Visualization, 2013 IEEE, pp. 299-306.
- [9] Bai X., White D., Sundaram D.: Context Adaptive Visualization for Effective Business Intelligence, Proceedings of ICCT2013, pp. 786-790
- [10] G140: Separating information and structure from presentation to enable different presentations, <http://www.w3.org/TR/WCAG-TECHS/G140.html>
- [11] Fowler M.: Development of Further Patterns of Enterprise Application Architecture, <http://martinfowler.com/eaDev>
- [12] TOGAF® Version 9.1, Open Group Standard, The Open Group, 2009-2011 pp. 692
- [13] The Open Group: ArchiMate 2.0 Specification (2009-2012), p. 183
- [14] ISO/IEC TR 9126-3:2003 Software engineering - Product quality - Part 3: Internal metrics, International Org. for Standardization (ISO)
- [15] ISO FDIS 9241-210:2009. Ergonomics of human system interaction - Part 210: Human-centered design for interactive systems, International Org. for Standardization (ISO)
- [16] Ch. Wang, H. Chen: From Data to Knowledge to Action: A Taxi Business Intelligence System, 15th International Conference on Information Fusion (FUSION), 2012, p. 1632 – 1628
- [17] J. Cooley, J. Petrusich, Delivering optimal real-time manufacturing intelligence, Proceedings of PICMET '13: Technology Management for Emerging Technologies, 2013, p. 1658 – 1668.
- [18] Ekaterina Kuts: Playful User Interfaces: Literature Review and Model for Analysis, Breaking New Ground: Innovation in Games, Play, Practice and Theory. Proceedings of DiGRA 2009
- [19] Janne Paavilainen: Designing Social Network Games with SoPlay Heuristics, MindTrek 2010 Conference Workshop, Tampere, Finland.
- [20] K. Jamróz, D. Pitulej, J. Werewka: Adapting Enterprise Architecture at a Software Development Company and the Resultant Benefits, in P. Avgeriou and U. Zdun (Eds.): ECSA 2014, LNCS 8627, pp. 170–185 (2014)
- [21] X. Dai, N. Matta, G. Ducellier: CKD: a Cooperative Knowledge Discovery Model for Design Project. Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, 1363–1369