Business Rules Design Method for Business Process Management

Grzegorz J. Nalepa
Institute of Automatics,
AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
Email: gjn@agh.edu.pl

Maria A. Mach
Institute of Business Informatics,
Wrocław University of Economics,
Komandorska 118/120, 53-345 Wrocław, Poland
Email: maria.mach@ue.wroc.pl

Abstract—**The paper deals with the problem of designing business rules (BR) for the business process modeling (BPM). It shows the role of knowledge in BPM and points out the issues that arise from the link between knowledge and the BPM. It presents the challenges of BR modeling, shows some disadvantages of existing and commonly used methods. As a solution, a method based on the XTT approach is outlined and discussed.**

Index Terms—**Knowledge Representation, XTT, HeKatE, Business Process Management, Business Rules**

I. KNOWLEDGE IN BUSINESS PROCESS MANAGEMENT

LET US begin with a definition of *Business Process Management* (BPM for short). In [1] we find the following one: "supporting business processes using methods, techniques and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information". This definition is of extremely high importance, because while analyzing it, one can immediately see the main issues or problems connected with BPM. These are the questions of incorporating human knowledge into BPM systems, knowledge representation, and mining knowledge from BPM systems.

Therefore, BPM is strictly linked with such areas of Artificial Intelligence (AI) as Knowledge Management (KM), Knowledge Engineering (KE) and Business Intelligence (BI). Knowledge is an essential factor in practical BPM. Knowledge related issues include: acquisition, representation, evaluation, and processing. Knowledge representation methods need proper syntax, visual representation, and formal foundations. These issues have been extensively studied in the field of Knowledge Engineering. Before applying any of the knowledge representation techniques, knowledge on processes has to be gathered and acquired both from existing systems and people. While the first task is not very complicated, the second one is not trivial. Most of employees' knowledge is a tacit one [2], not easy to be expressed. It may nevertheless not be omitted, as knowledge and business processes are integrated and should be evaluated as a whole. Processes involve collaboration between individuals and/or groups to achieve a goal [3] – in BPM there is a need for recognizing the involvement of humans in the execution of business processes

to properly model them. Business processes can be therefore seen as social constructs [3].

The above remarks show the link between BPM and KM. There is also an approach to support users in managing process execution quality. This issue forms a link between BPM and BI. Analyzing logging information about business processes may provide important knowledge on high and low-quality processes in the past. This knowledge may be used for explanation and for predicting potential problems [4]. Knowledge engineering is strictly linked with the above issues.

In AI, *rules* are probably the most popular choice for building knowledge-based systems (KBS), that is the so-called rule-based expert systems [5], [6]. Rule-based systems (RBS) are used extensively in practical applications, especially in domains such as automatic control, decision support, and system diagnosis. They constitute today one of the most important classes of KBS. Practical construction of a rule-based knowledge base, also referred to as the *rulebase* involves number of important steps. These include rule attribute specification, rules design, rulebase evaluation, possibly a practical implementation using an inference mechanism.

Recently, a new approach to practical knowledge representation based on rules, has been gaining popularity. This is the so-called *Business Rules Approach* (BR). As stated in [7], "a business rule is a statement that defines or constraints some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business".

The focus of the paper is on the design and modeling of BR in the BPM (see Sect. II). The paper is dedicated to the discussion of the most important issues concerning practical design of business rules (see Sect. III), found in the BPM. In the paper a new design method is described, using an example business rulebase described in Sect. IV. This example has been originally designed using the BPMN (Business Process Modeling Notation) [8], and accompanied with business rules. The method presented in this paper, is centered around the XTT approach, developed within the HeKatE project presented in Sect. V. In Sect. VI it is discussed, how applications of these methods could improve aspects of BPM. The paper ends with concluding remarks in Sect. VII.

II. Business Process Modelling

The main aims of *business process modeling* are as follows: *descriptive* – what happens during a business process? In what way the process has been performed? What improvements have to be made?; *prescriptive* – allows for a definition of a business process and how a process should be performed, it lays down rules, guidelines and behavior patterns; *explanatory* – links processes with the requirements that are to be fulfilled, explains the rationale of business processes.

These aims lead to the formulation of the requirements that a business process model has to fulfill. First of all, a model has to provide a holistic approach dealing with organizational and technical issues [9]. Next, BP models should have a strong formal foundation. It is so because formal models: a) are unambiguous, b) increase the potential for analysis [1].

These are the basic requirements for a good BP model. Other may be formulated as well; e.g. according to [7] and [10]) a BP model should possess the following features

- unified model execution and manipulation,
- state management,
- time-based exception management,
- robust process monitoring and analysis,
- nested model support,
- concurrent model support,
- standards based,
- high scalability and reliability,
- explicit expression of business rules,
- coherent representation,
- evolutionary extension,
- declarative nature.

There are several techniques for BP model specifications. Some are based on Petri nets [1], others use the UML notation. It must be nevertheless pointed out that UML, although widely used and adopted, is not designed to map to business execution languages. Two important new approaches to business process modeling are the *Business Rules Project* [7] and the *Business Process Modeling Notation* [8], [11].

The BPMN is presented by its authors as a new standard for modeling both business processes and web service processes. It consists of one diagram, the so-called Business Process Diagram (BPD), composed of several groups of visual elements, namely: Activities, Events, Gateways, Connections, Artifacts, and Swimlanes. As the authors of the technique claim, BPD is easy to use and understand, allows to model very complex business processes, and can be naturally mapped to business execution languages. Nevertheless it is not free from disadvantages whet it comes to a practical application.

The most important limitation of this method is its high abstraction level, which makes creating an execution environment for BPMN challanging. In most cases the process are mapped in the *Service-Oriented Architecture* (SOA) [12], [13], with low-level execution delegated to third party environmnts. However, this approch makes formalization and evaluation of such a transition from conceptual process specification to the executable form very difficult.

III. Business Rules Approach

A. Business Rules Concepts

Business Rules (BR) approach [14], [15] is based on concepts borrowed from knowledge engineering and rule-based systems. It is becoming an important approach in business application development, especially on the Java platform.

A classic description of the main principles of the approach is given in [14]. According to it, rules should be: written and made explicit, expressed in plain language, motivated by identifiable and important business factors, single sourced, specified directly by people who have relevant knowledge, managed, and built on facts. Rules should also exist independent of procedures and workflows.

There are number of *rule types* identified in the BR approach, such as: reactive rules (event-condition-action rules), transformation rules (functional-equational rules), derivation rules (implicational-inference rules), also ones restricted to facts ("premiseless" derivation rules) and queries ("conclusionless" derivation rules), as well as integrity-constraints (consistency-maintenance rules).

Business rules design uses some established *visual representations*. Depending on the design approach these are some classic tools such as simple propositional decision tables, or some high-level conceptual tools such as URML [16]. There are attempts to officially define main aspects of the approach. A good example is the *emantics of Business Vocabulary a Business Rules Specification* (SBVR), see [17].

From the point of view of formal knowledge engineering, some major issues can be pointed out. They are related to: a) logical foundations, b) visual representation, and c) formal analysis and verification of BR systems.

The *first problem* concerns the *logical foundations* of BR systems. From a point of view of classical KE, a rule-based expert system consists of a knowledge base and an inference engine. The KE process aims at designing and evaluating the knowledge base, and implementing a proper inference engine. The process of building the knowledge base involves the selection of a knowledge representation method, knowledge acquisition, and possibly low-level knowledge encoding. In order to create an inference engine a reasoning technique must be selected, and the engine has to be programmed. In the formal analysis of RBS [6] some important aspects of the design and implementation are identified:

1) rulebase design, including: the formal logical language of the representation, formal syntax of the representation method, representation expressiveness, which is often a function of the expressiveness of the underlying logic, and particular rule syntax.
2) inference engine implementation, including: inference strategy, interpreter model, including rule matching method, conflict resolution algorithm.

Unfortunately it can be observed, that common approaches to BR tend to mix these formal aspects. The concept of "business rules types" is both misleading and imprecise. A proper formal analysis of BR should provide a more adequate classification.

The *second problem* is related to the *visual representation* used in the design of BR systems. Visual representations used, have scalability problems (it's easy to draw diagrams of several rules, but it becomes very difficult to cope with tens of rules). Lack of well-defined formal foundations of these representations leads to problems with automatic transformation of visual model to the logical one.

The *third problem* concerns the *formal analysis end verification* of BR systems. As the number of rules exceeds even relatively very low quantities, it is hard to keep the rule-base consistent, complete, and correct. These problems are related to knowledge-base verification, validation, and testing. The selection of appropriate software tools and programming languages is non-trivial either.

These issues are very rarely considered in the BR design. It seems that analysis (where issues such as verification, validation, and evaluation are even not properly separated) is simply considered testing. So the analysis of the *knowledge base* is implicitly substituted by testing of the *implementation*. However, in the KE approach, a proper analysis of the knowledge base minimizes the need for testing.

B. Computer Tools for Business Rules

There has been a very active development of computer tools for BR in recent years. Today, there are number of BR-related solutions available. For overview of these tools see [18], [19]. While these developments aim at improving the design process of BR applications, it seems that they fail to provide effective solutions. There seem to be two main problems.

The first one is unsuitable knowledge representation used during the design. The basic representation, such as simple decision tables, is then used in a design tool which is often inefficient (such as Excel). What is even more important is the fact, that these tools do not try to overcome the so-called *semantic gap* [20] between declarative design and procedural implementation. Tools like Drools (www.drools.org) provide a kind of rule meta-language embedded in the implementation language (e.g. Java). This does not seem to be an efficient solution, since it mixes two different semantics.

The second one concerns analysis and verification of BR. It is bizarre that there are virtually no specialized evaluation tools. Some simple testing or syntax checking features during the design (which are not even present in case of Excel-based solutions) are a step backwards, compared to evaluation and analysis tools developed several years ago (e.g. see [21]).

IV. PRACTICAL BUSINESS RULEBASE DESIGN EXAMPLE

Let us consider a classic illustrative BR example. It has been presented on the Business Rules Forum in 2005 [22]. The example concerns the UServ Financial Services Company, which provides a full service portfolio of financial products, including: Insurance, and Banking. UServ plays a balancing act between rewarding their best clients and managing the risk inherent in providing on-going service to clients whose portfolios are profitable, but violate the eligibility rules of individual products. The business rules address eligibility,

pricing and cancellation policies at both the individual product and portfolio level. The case study [22] focuses on UServ's vehicle insurance products, but differentiates the basic business rules from those that apply to preferred and elite clients. In the BR rulebase three groups of rules are identified: Client Segmentation Business Rules, Eligibility Business Rules, and Pricing Business Rules.
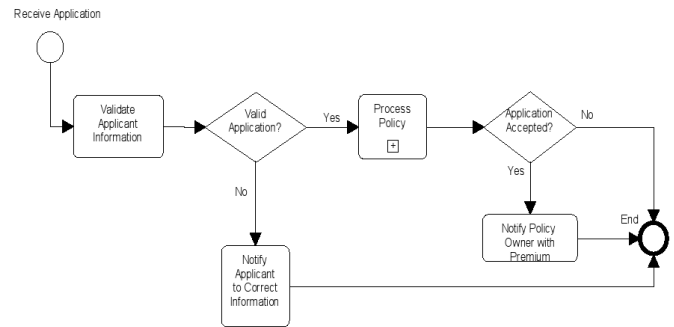

Fig. 1. The BPM of Application Processing

The practical design of the system presented in [22] has been carried out using the *Business Process Modeling Notation* [8] (BPMN). This is the main notation used in the BPM. The main process of the vehicle insurance policy processing can be observed in Fig. 1. In this process a subprocess "Process Policy" is used. It uses the scoring of the policy delegated to a subprocess shown in Fig. 2.
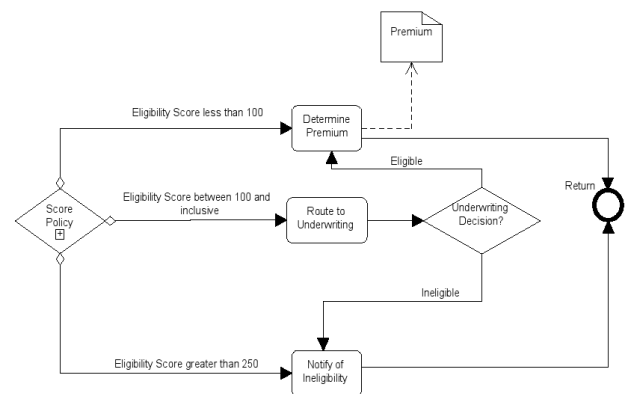

Fig. 2. The BPM of Policy Scoring

This model is accompanied with business rules specific to the activities. Every activity in the diagram includes number of rules. For example „Driver Eligibility Rules" include the assessment of the „Driver Age Category":

```
If the driver is male and is under the age of 25,
    then young driver.
If the driver is female and is under the age of 20,
    then young driver.
If young driver and driver has training certification,
    then eligible driver.
If the driver is over the age of 70,
    then senior driver.
If senior driver and driver has training certification,
    then eligible driver.
```

In this case the BR approach is being integrated on a low-level with use of rules as the way of expressing activities in given contexts. The main structure of the process is modeled using the BPMN.

In the following section an alternative way of integration is proposed. It consists in using an advanced rule design and implementation process. The process is centered around the XTT visual and logical design method for RBS. XTT has a rich semantics, which allows for designing number of systems, including business-related. The process is transparent and coherent, offering design methods that allow for both high-level conceptual design, as well as low-level rule implementation.

V. The XTT Approach

The *HeKatE* project (see hekate.ia.agh.edu.pl) aims at providing an integrated methodology for the design, implementation, and analysis of rule-based systems [23], [6]. The main principles of the project are based on a critical analysis of the state-of-the art of the rule-based systems design, see [24]. These are: Formal Language for Knowledge Representation, Internal Knowledge Base Structure, and Systematic Hierarchical Design Procedure. A complete, well-founded design process that covers all of the main phases of the system lifecycle, from the initial conceptual design, through the logical formulation, all the way to the physical implementation is proposed. A constant verification of the model w.r.t. critical formal properties, such as completeness is provided.

HeKatE introduces a formalized language for rule representation [24]. Instead of simple propositional formulas, the language uses expressions in the so-called *attributive logic* [6]. This calculus has a stronger expressiveness than the propositional logic, while providing tractable inference procedures for extended decision tables. The current version of the rule language is called XTT^2 [25]. The current version of the logic, adopted for the XTT^2 language, is called ALSV(FD) (*Attributive Logic with Set Values over Finite Domains*).

Based on the logic, a rule language called XTT is provided [26], [25]. XTT stands for *eXtended Tabular Trees*. The language is focused not only on providing an extended syntax for single rules, but also allows for an explicit structurization of the rule base. This solution allows for identifying system contexts during the rule base design. XTT introduces explicit inference control solutions, allowing for a fine grained and more optimized rule inference than in the classic Rete-like solutions. XTT has been introduced with the visual design support in mind. The representation has a compact and transparent visual representation suitable for visual editors.

HeKatE also provides a complete hierarchical design process for the creation of the XTT-based rules. The main phase of the XTT rule design is called the *logical design*. The logical rule design process may be supported by a preceding *conceptual design* phase. In this phase the rule prototypes are built with the use of the so-called *Attribute Relationship Diagrams*. The ARD method has been introduced in [27], and later refined in [6]. The principal idea is to build a graph, modelling functional dependencies between attributes on which the XTT rules are built. The version used in HeKatE is called ARD+ as discussed in [28]. The practical implementation on the XTT rule base is performed in the physical design phase. In this stage the visual XTT model is transformed into an algebraic presentation syntax called HMR. A custom inference engine, HeaRT runs the XTT model described in HMR.

The HeKatE design process is supported by a set of tools called HaDEs. They support the visual design and the automated implementation of rule-based systems (see https://ai.ia.agh.edu.pl/wiki/hekate:hades). *HJEd* visual editor supports the ARD design process implemented in Java. *VARDA* is a prototype semi-visual editor for the ARD diagrams implemented in Prolog, with an on-line model visualization with Graphviz. Once created, the ARD model can be saved in a XML-based HML (HeKatE Markup Language) file. The file can be then imported by the HQEd design tools supporting the logical design. *HQEd* provides support for the logical design with XTT. In the figure some additional decision tables to input attribute values are present. It is able to import a HML file with the ARD model and generate the XTT prototype. The editor is integrated with a custom inference engine for XTT^2 called HeaRT. *HeaRT* (HeKatE Run Time) is a dedicated inference engine for the XTT^2 rule bases. It is implemented in Prolog in order to directly interpret the HMR representation which is generated by HQEd. HMR (HeKatE Meta Representation) is a textual representation of the XTT^2 logic designed by HQEd. *HalVA* (HeKatE Verification and Analysis) is a modularized verification framework provided by HeaRT. So far several plugins are available, including completeness, determinism and redundancy checks. The plugins can be run form the interpreter or from HQEd using the communication protocol.

Let us now show, how the USery example can be designed using this approach.

VI. Visual Business Rules Design with XTT

The conceptual design with ARD [27] is based on the idea of the full input/output specification of the object attributes involved in the business process. In this case these would be: policy (for the car), client (that buys the policy), score and premium. In the this design phase a complete specification of system attributes has been formulated. The complete design diagram can be observed in Fig. 3. ARD provides a hierarchical model. At the top-most *Level 0*, the most basic relation between some general input and output attributes is visualized. At every subsequent level, this relation is becoming more specific. At the last level the diagram gives full specification of relations between physical system attributes. For example the level 2 diagram (third from the top) means: „Client, Policy and Premium depend on the PolicyScore attribute". In the next level this attribute is in fact specified to three more detailed attributes. For more details on syntax and semantics of ARD see [27], [28].

During the ARD design, two classes of attributes are used: *conceptual* and *physical*. The so-called *conceptual* attributes, or the *generalized* attributes, get specified, during the conceptual design, into the physical ones, present in the final rules.
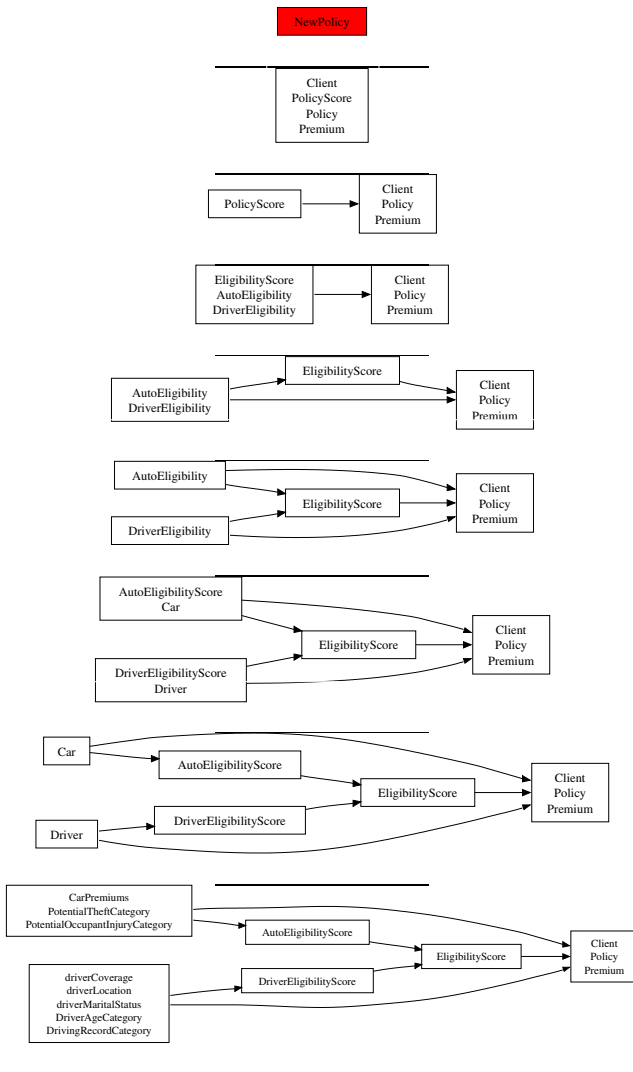
Fig. 3. ARD Conceptual Design for the XTT model

The *physical* attribute specification in the XTT/ARD contains:

- attribute names e.g. `aPotentialOccupantInjuryRating`,
- abbreviated attribute names, more suitable for compact Prolog implementation, e.g. `aPOIRt`,
- attribute types, e.g. `symbolic`, and
- specification of attribute value domains, e.g. `[Low,Moderate,High]`.

The last level of the ARD is a *table scheme specification* for the XTT logical design method. During this phase, rules are built using specific attribute values. Using the results of the conceptual analysis, the design of the rule base is put forward, using the XTT representation method [29], [30]. In this method the rule base is visualized using tables grouping rules operating in the same context (on the same attributes) into a tree-like structure. The compact representation is based on a expressive *attributive* language, with use of *non-atomic* values.

The XTT logical design involves using the bottom level of ARD design, as a base for tables containing rules. The lowest ARD level includes all of the physical attributes present in rules. In a general case, the inference process is non-monotonic, since XTT allows for dynamic modification of the knowledge base. The XTT structure corresponds to the structure of the decision process involved in the business process. XTT is automatically transformed into an executable Prolog-based representation [29], [30].

The automatic transformation of XTT to Prolog, which can be done at any stage of the logical design, allows for an *on-line* evaluation of the rule base. HeaRT provides a number of Prolog-based verification plugins. They verify some important formal properties of the system, such as redundancy, completeness, or determinism. What is important in this case, is the possibility of *formal* analysis, *during* the design. The details of the evaluation procedure are out of scope of this paper. They have been presented in [31], [30].

Compared to the BPMN representation and design, the XTT/ARD model is complete and transparent. It offers a hierarchical representation of the whole knowledge base. While the ARD could be compared to BPMN design, the XTT allows for direct rule modeling, and the Prolog transformation gives an executable prototype. Both BPMN and the XTT approaches have strong formal foundations. Yet some features of the latter solution make it a more advanced approach to the BPM. These include automatic translation of the design specification into a low-level code, a well-defined system semantics and the possibility of on-line system verification.

VII. CONCLUDING REMARKS

The goal of any business process management solution is to build a better business. Process design and automation technologies are being increasingly used to improve quality and efficiency of business processes, and – in consequence – to deliver services rapidly and reliably. As any business process is built (sometimes implicitly) of business rules, the problem of designing and modeling the business rules is the core question in the BPM. Many solutions to this problem have been proposed and used so far, each them having both advantages and disadvantages. In the paper, the role of knowledge in the BPM, which leads to the question of proper knowledge representation in the form of business rules, is pointed out.

The Business Rules Approach was outlined, and the main disadvantages of common approaches to BR design and modeling were pointed out, as well as the most important problems to be solved. As a solution to these problems, the XTT approach was proposed. Broadly speaking, this approach fulfills most of the requirements for BR design and modeling tools, that were pointed out in Section II. In particular, it allows for a direct model execution and manipulation, and provides a coherent representation of rules, yielding a single formal representation of all aspects of an enterprise's structure and operations. The possibility of on-line evaluation of the rule base, provided by the XTT approach assures correctness of the system being built. XTT is developed within the *HeKatE* project (see hekate.ia.agh.edu.pl) that aims at providing an

integrated methodology for the design, implementation, and analysis of rule-based systems.

Keeping in mind these important features of the proposed approach, it may be then claimed that the XTT solution improves a very important aspect of the BPM – namely the design and modeling of BR, leading to a complete, transparent and elegant representation of the rule base. As it has been pointed out, the more proper and accurate knowledge representation, the more possibilities for the BPM. Some issues that were not addressed in this paper still need to be elaborated. These are e.g. knowledge evaluation questions and knowledge mining, as well es extended design of the rule-based business processes. Future research directions are therefore twofold: first, to focus on the on-line evaluation of the rule base in the process of BR design, second, comparing the XTT and BPMN approaches as a source for knowledge discovery.

REFERENCES

[1] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business process management: A survey." in *Proc. Business Process Management: International Conference, BPM 2003, June 26-27, Eindhoven, the Netherlands*, ser. LNCS. Springer, 2003, pp. 1–12.

[2] N. Gronau, C. Mueller, and M. Uslar, "The kmdl knowlede management approach: Integrating knowledge conversions and business process modeling," in *Proc. 5th International conference on practical aspects of knowledge management Vienna, Austria, 02.12. 2004*, ser. LNCS, vol. 3336. Springer, 2004, pp. 1–10.

[3] A. Lindsay, D. Dawns, and K. Lunn, "Business processes - attempts to find a definition," *Information and Software Technology*, vol. 45, no. 15, pp. 1015–1019, December 2003, elsevier.

[4] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M. C. Schan, "Business process intelligence," *Computers in Industry*, vol. 53, no. 3, pp. 321–243, April 2004, elsevier.

[5] P. Jackson, *Introduction to Expert Systems*, 3rd ed. Addison–Wesley, 1999, iSBN 0-201-87686-8.

[6] A. Ligęza, *Logical Foundations for Rule-Based Systems*. Berlin, Heidelberg: Springer-Verlag, 2006.

[7] D. Hay, A. Kolber, and K. A. Healy, "Defining business rules - what they really are. final report." Business Rules Group, Tech. Rep., July 2000.

[8] OMG, "Business process modeling notation (bpmn) specification," Object Management Group, Tech. Rep. dtc/06-02-01, February 2006.

[9] D. Karagiannis, "Bpms: Business process management systems," *SIGOIS Bulletin*, vol. 16, no. 1, pp. 10–13, August 1995.

[10] T. McDaniel, "Ten pillars of business process management," *eAI Journal*, pp. 30–34, November 2001.

[11] M. Owen and J. Raj, "Bpmn and business process management. introduction to the new business process modeling standard." OMG, Tech. Rep., 2006.

[12] T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

[13] K. Pant and M. Juric, *Business Process Driven SOA using BPMN and BPEL: From Business Process Modeling to Orchestration and Service Oriented Architecture*. Packt Publishing, 2008.

[14] R. G. Ross, *Principles of the Business Rule Approach*, 1st ed. Addison-Wesley Professional, 2003.

[15] B. von Halle, *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. Wiley, 2001.

[16] S. Lukichev and G. Wagner, "Visual rules modeling," in *Sixth International Andrei Ershov Memorial Conference PERSPECTIVES OF SYSTEM INFORMATICS, Novosibirsk, Russia, June 2006*, ser. LNCS. Springer, 2005.

[17] OMG, "Semantics of business vocabulary and business rules (sbvr)," Object Management Group, Tech. Rep. dtc/06-03-02, 2006.

[18] G. J. Nalepa, "Rule-based systems design and implementation : methodologies and technologies," in *CMS'05. Plenary lectures and special session papers : Computer Methods and Systems*, M. S. Ryszard Tadeusiewicz, Antoni Ligęza, Ed., vol. 1, AGH University of Science and Technology Cracow, Jagiellonian University, Cracow University of Technology. Kraków, Poland: Oprogramowanie Naukowo-Techniczne, November 2005, pp. 329–340.

[19] G. J. Nalepa and A. Ligęza, "New generation of expert systems development tools," in *Inżynieria Wiedzy i Systemy Ekspertowe*, A. Grzech, Ed., vol. 2. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2006, pp. 63–72.

[20] D. Merrit, "Best practices for rule-based application development," *Microsoft Architects JOURNAL*, vol. 1, 2004.

[21] R. Gerrits and S. Spreeuwenberg, "Valens: A knowledge based tool to validate and verify an aion knowledge base." in *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany*, 2000, pp. 731–738.

[22] BRForum, "Userv product derby case study," Business Rules Forum, Tech. Rep., 2005.

[23] J. Giarratano and G. Riley, *Expert Systems. Principles and Programming*, 4th ed. Boston, MA, United States: Thomson Course Technology, 2005, iSBN 0-534-38447-1.

[24] G. J. Nalepa and A. Ligęza, "Hekate methodology, hybrid engineering of intelligent systems," *International Journal of Applied Mathematics and Computer Science*, 2009, accepted for publication.

[25] ——, "Xtt+ rule design using the alsv(fd)," in *ECAI 2008: 18th European Conference on Artificial Intelligence: 2nd East European Workshop on Rule-based applications, RuleApps2008: Patras, 22 July 2008*, A. Giurca, A. Analyti, and G. Wagner, Eds. Patras: University of Patras, 2008, pp. 11–15.

[26] ——, "A graphical tabular model for rule-based logic programming and verification," *Systems Science*, vol. 31, no. 2, pp. 89–95, 2005.

[27] ——, "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.

[28] 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.

[29] G. J. Nalepa, "Meta-level approach to integrated process of design and implementation of rule-based systems," Ph.D. dissertation, AGH University of Science and Technology, AGH Institute of Automatics, Cracow, Poland, September 2004.

[30] 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.

[31] A. Ligęza and G. J. Nalepa, "Visual design and on-line verification of tabular rule-based systems with xtt," in *Marktplatz Internet: Von e-Learning bis e-Payment : 13. Leipziger Informatik-Tage, LIT 2005*, ser. Lecture Notes in Informatics (LNI), K. P. Jantke, K.-P. Fähnrich, and W. S. Wittig, Eds. Bonn: Gesellschaft fur Informatik, 2005, pp. 303–312.