

Proposal of Business Process and Rules Modeling with the XTT Method*

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

Abstract

The paper deals with the problem of modeling business processes and business rules (BR). Business Rules are becoming an important solution for knowledge modeling. The paper presents the challenges of BR modeling, shows some disadvantages of existing and commonly used methods, such as the BPMN. As a solution, an approach based on the XTT (Extended Tabular Trees) expert system design and implementation method is outlined and discussed. It improves the design and modeling of BR, leading to a more complete, and transparent representation of the rule base.

1. Knowledge in Business Processes

Knowledge is an essential factor in practical Business Process Management (BPM) [19]. Knowledge related issues include: acquisition, representation, evaluation, and processing. 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 acquired 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 [3], 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 a collaboration between individuals and/or groups to achieve a goal [8] – in the BPM there is a need for recognizing the involvement of humans in the execution of business processes to properly model them.

In AI, *rules* are probably the most popular choice for building knowledge-based systems (KBS), that is the so-called rule-based expert systems [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.

Recently, a new approach to practical knowledge representation based on rules, has been gaining popularity. This is the so-called *Business Rules Approach* (BR), see [18, 20]. As stated in [4], "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 (Sect. 2). The paper is dedicated to the discussion of the important issues concerning practical design of business rules (see Sect. 3), found in the BPM. In the paper a new design method is described, using a classic business rulebase example described in Sect. 4. This example has been originally designed using the BPMN (Business Process Modeling Notation) [16], and accompanied with business rules. The method presented in this paper, is centered around the XTT, an expert system design method, presented in Sect. 5. In Sect. 6 it is discussed, how applications of these methods could improve selected aspects of the BPM. The paper ends with future work (Sect. 7) and concluding remarks in Sect. 8.

2. Business Process Modeling

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

These lead to the formulation of the business process (BP) model requirements. First of all, the model has to provide a holistic approach dealing with organizational

*The paper is supported by the HEKATE Project funded from 2007–2009 resources for science as a research project.

and technical issues [5]. Next, BP models should have a strong formal foundation. It is so because formal models are unambiguous, and increase the potential for analysis [19]. These are the basic requirements for a good BP model. Other requirements may be formulated as well. For example a BP model should possess the following features ([4, 10]): unified model execution and manipulation, state management, time-based exception management, robust process monitoring and analysis, nested model support, concurrent model support, explicit expression of business rules, coherent representation, declarative nature.

There are several techniques for BP model specifications, such as Petri nets based [19]. Some make use of the UML notation. UML however, although widely used and adapted, is not expressly designed to map to business execution languages. A new formal approach to business process modeling, is the *Business Process Modeling Notation* (BPMN) [16]. The BPMN is presented as a new standard for modeling business processes. BPMN consists of one diagram - the so-called Business Process Diagram (BPD). It consists of several groups of visual elements, namely: Activities, Events, Gateways, Connections, Artifacts, and Swimlanes. BPD is supposed to be 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, briefly discussed in Sect. 3.

The BPMN is accompanied with the *Semantics of Business Vocabulary a Business Rules Specification* [17], which aims at modeling the business concepts. It also supports another approach used in practical BP modeling, that is the one proposed by the *Business Rules Project* [4].

3. Business Rules Approach

Business Rules (BR) approach [18, 20] 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 [18]. According to it, rules should be: written and made explicit, motivated by important business factors, managed, and built on facts, and facts should build on concepts as represented by terms.

There are number of *rule types* identified in the BR approach, such as: reactive rules (event-condition-action 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 de-

cision tables, or some high-level conceptual tools, such as URML [9], developed by the REVERSE project (see reverse.net). It provides a novel application of UML-based design to business rules modeling. The model can be easily integrated with object-oriented applications.

From the point of view of formal knowledge engineering, some major issues concerning BR modeling 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. In the formal analysis of RBS [6] important aspects of the design and implementation of the RBS are identified: *rulebase design*, including: a formal logical language syntax of the representation, representation expressiveness, which is often a function of the expressiveness of the underlying logic, and particular rule syntax. *Inference engine implementation*, including: inference strategy, interpreter model, and conflict resolution algorithm. Unfortunately, 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 should provide a more adequate classification of BR.

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 rarely considered in the BR design. Analysis is often considered testing. So the analysis of the *knowledge base* is implicitly substituted by testing of the *implementation*. While, in the KE approach, a proper analysis of the knowledge base minimizes the need for testing.

There has been a very active development of computer tools for BR in recent years. Today, there are number of BR-related solutions available. 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 the unsuitable knowledge representation used during the design. The basic representation, such as simple decision tables, is often inefficient. What is impor-

tant is the fact, that these tools do not try to overcome the so-called *semantic gap* between declarative design and procedural implementation.

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 are a step backwards, compared to some rule-based evaluation and analysis tools developed several years ago (e.g. see [2]). In fact the tool presented in [2], now developed by *LibRT* (www.librt.com), is one of the few specialized rule analysis tools now available.

4. Practical Business Rulebase Design Example

Let us consider a classic illustrative BR example. It has been presented on the Business Rules Forum in 2005 [1]. The example concerns the UServ Financial Services Company, which provides a full service portfolio of financial products, including: Insurance, and Banking. UServ balances between rewarding their best clients and managing the risk inherent in providing service to clients whose portfolios are profitable, but violate the eligibility rules of individual products.

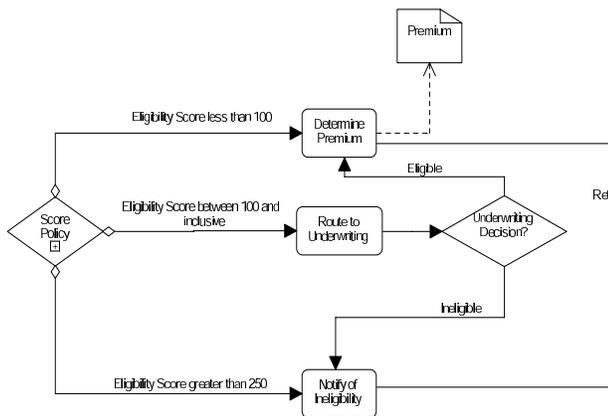


Figure 1. The BPM of Policy Processing.

UServ’s business rules are an essential component for managing this risk. The business rules address eligibility, pricing and cancellation policies at both the individual product and portfolio level. The case study [1] 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, Eligibility, and Pricing Business Rules.

The practical design of the system presented in [1] has been carried out using the *Business Process Modeling Notation* [16] (BPMN). This is the main notation used in the BPM (see also Section 2). The main subprocess of the ve-

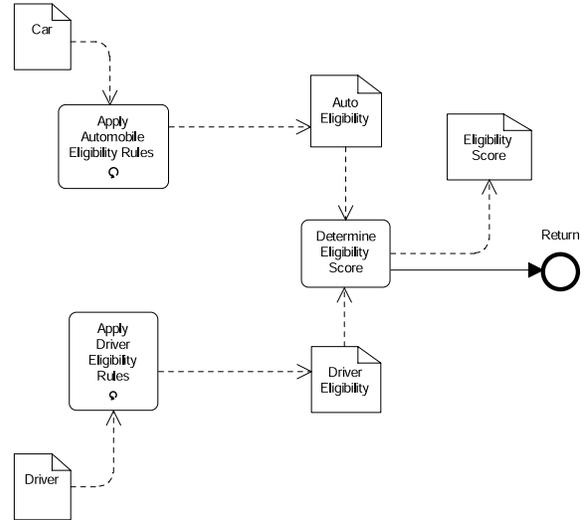


Figure 2. The BPM of Policy Scoring.

hicle insurance policy processing can be observed in Fig. 1. In this process the scoring of the policy is delegated to a subprocess shown in Fig. 2.

This model is accompanied with activity-specific business rules. Every activity in the diagram includes number of rules. For example „Automobile Eligibility Rules” include the assessment of the „Potential Occupant Injury Category”:

- If the car has no airbags, then the car’s potential occupant injury rating is extremely high.
- If the car only has driver’s air bag, then the car’s potential occupant injury rating is high.
- If the car has driver’s and front passenger air bags, then the car’s potential occupant injury rating is moderate.
- If the car has driver’s front passenger and side panel air bags, then the car’s potential occupant injury is low.
- If the car is a convertible and has no roll bar, then the potential occupant injury is extremely high.

So in this case, the main structure of the business process is modeled using the BPMN, and then the BR approach is being integrated on a low-level with use of rules as the way of expressing activities in the given contexts of the process. 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. XTT has a rich semantics, which allows for designing number of systems, including business-related. The process is transparent and coherent, offering method that allows for both the high-level conceptual design, as well as the low-level rule implementation.

5. The XTT Approach

The integrated design process discussed here can be considered a *top-down hierarchical design methodology*, based on the idea of meta-level approach to the design process. It

includes three phases: conceptual, logical, and physical. It provides a clear separation of logical and physical (implementation) design phases. It offers equivalence of logical design specification and prototype implementation, and employs XTT [13] (*Extended Tabular-Trees*), a hybrid knowledge representation.

The main goal of the methodology is to move the design procedure to a logical level, where knowledge specification is based on the use of abstract rule representation. The design can be automatically translated into a low-level Prolog code. On the other hand, selected formal system properties can be automatically analyzed on-line during the design, so that its characteristics are preserved. The generated Prolog code constitutes a prototype implementation of the system. Since it is equivalent to the visual design specification it can be considered an executable specification.

The main idea behind the XTT representation and design method aims at combining selected existing approaches, namely decision trees and decision tables, by building a special hierarchy of Object-Attribute-Tables [7, 6]. It allows for a hierarchical visual representation of the OAT tables linked into tree-like structure, according to the control specification provided. XTT as a design and knowledge representation method offers transparent, high density knowledge representation as well as a formally defined logical, Prolog-based interpretation, while preserving flexibility with respect to knowledge manipulation.

The *conceptual design* of the RBS aims at modeling the most important features of the system, i.e. attributes and functional dependencies among them. ARD (*Attribute-Relationship Diagrams* [12, 6]) allows for specification of functional dependencies of system attributes using a visual representation. An ARD *diagram* is a conceptual system model at a certain abstract level. It is composed of one or several ARD *tables*. A partial order relation among the tables is represented with *arcs*. The ARD model is also hierarchical. The most abstract level 0 diagram shows the functional dependency of *input* and *output* attributes. Lower level diagrams are less abstract, closer to full specification.

Using XTT as a core, an *integrated design process*, covering the following three phases is considered: *Conceptual modeling*, in which system attributes and their functional relationships are identified; during this design phase the ARD modeling method is used; *Logical design with on-line verification*, during which system structure is represented as XTT hierarchy, which can be instantly analyzed, verified and even optimized on-line, using Prolog; *Physical design*, in which a preliminary Prolog-based *implementation* is carried out. A RuleML translation of the XTT rule base has been also proposed.

A prototype CASE tool for the XTT called *Mirella Designer* has been developed. It supports XTT-based visual design methodology, with an integrated, incremental design

and implementation process, providing the possibility of the on-line, incremental, verification of formal properties. It offers a direct translation to a formal, Prolog-based representation equivalent to the XTT knowledge base [14], with an *on-line* knowledge analysis, including formal verification, and refinement of the knowledge base [14].

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

6. Visual Business Rules Design with XTT

The conceptual design with ARD 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), driver (of the car), the car itself. 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. In this case, the level 0 diagram states, that „*policy pricing* depends on some features of the *client*, *driver*, and the *car*.” At every subsequent level, this relation is becoming more specific. At the last, bottom level, the diagram gives full specification of relations between physical system attributes. For more details on syntax and semantics of ARD see [12].

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. 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, Mod, 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 actual design of the rule base is put forward, using the XTT representation method. In this method the rule base is visualized using a tables grouping rules operating in the same context (on the same attributes) into a tree-like structure. What is important, the representation is based on an expressive *attributive* language, with use of *non-atomic* attribute values. This allows for a more compact representation.

On the 6th level of ARD diagram shown in Fig. 3, all of the physical attributes are identified. They will be present in XTT rules. The full discussion of the example is long. Let us consider the „Automobile Eligibility Rules” that allow

number of Prolog-based verification plugins are provided. 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, see [14] for details.

Compared to the BPMN representation and design, the XTT/ARD model is more complete and transparent, offering a hierarchical representation. While the ARD could be compared to BPMN design, XTT allows for direct rule modeling, and the Prolog transformation gives an executable prototype. Both BPMN/BR and the ARD/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. On the other hand XTT itself is a generic rule design and implementation method, originally not targeted at modeling business processes (especially ARD semantics is far from state-based process semantics). This makes the application of XTT in this area non always trivial.

| (?) aCrAB | (?) aCrTp | (?) aCrRB | (+) aPOIRT |
|----------------------------|---------------|-----------|------------------|
| = None | = <any> | = <any> | = extremely high |
| = Driver | = <any> | = <any> | = high |
| In Driver,FrontPassenger | = <any> | = <any> | = aPOIRT |
| = FrontPassenger,SidePanel | = <any> | = <any> | = low |
| = <any> | = convertible | = no | = extremely high |

Table id. 1 - Automobile Eligibility

Figure 4. Potential Occupant Injury XTT Table

7. Future Work

The idea to design BPMN/BR systems with the XTT is just one area of application of the method. The development of the method itself continues within the HeKatE project, hekate.ia.agh.edu.pl, that aims at integrating selected AI methods, namely rule-based systems into the practical design and analysis of business software. The AI methods can in some cases improve practical design of such software, by providing formalized and expressive knowledge representation methods.

Future work on the XTT includes:

- new design tools, such as the HQEd structural editor, see screenshot in Fig. 4,
- integrations with other rule design languages (including RuleML) through RIF (see www.w3.org/2005/rules)
- pure XSLT transformations from the native XTTML into other XML formats, such as RuleML and RIF,
- extended business rules modeling (see [11]), as well as general business software modeling (see [15]),
- XTT diagram serialization in UML, with a translation to the XMI format, and
- formalism extensions, such as fuzzy and temporal rules.

In important area of development includes integration with some well established rule engines, such as *Drools*. It could be accomplished by providing a RIF compatible (see <http://www.w3.org/2005/rules/>) markup encoding for the XTT rules. Such a translation is currently in the works.

8. Concluding Remarks

Process design 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.

The Business Rules approach with the BPMN offer a way of rule modeling in business process. In the paper the main disadvantages of common approaches to BR design were pointed out. As a solution to these problems, the XTT method was proposed. XTT is a rule design, analysis, and implementation method, providing visual design process, with strong formal foundations. The approach aims at providing a coherent representation of rules. The possibility of on-line evaluation of the rule base, provided by the XTT approach assures correctness of the system being built. The solution improves an important aspect of the BPM – namely the design and modeling of BR, leading to a complete, and transparent representation of the rule base.

Acknowledgements The author wishes to thank Dr Maria Antonina Mach (maria.mach@ae.wroc.pl) from University of Economics in Wrocław for her support in writing the first version of this paper, as well as valuable remarks in improving the final one.

References

- [1] BRForum. Userv product derby case study. Technical report, Business Rules Forum, 2005.
- [2] 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*, pages 731–738, 2000.
- [3] N. Gronau, C. Mueller, and M. Uslar. The kmdl knowledge 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*, volume 3336 of *LNCS*, pages 1–10. Springer, 2004.
- [4] D. Hay, A. Kolber, and K. A. Healy. Defining business rules - what they really are. final report. Technical report, Business Rules Group, July 2000.
- [5] D. Karagiannis. Bpms: Business process management systems. *SIGOIS Bulletin*, 16(1):10–13, August 1995.
- [6] A. Ligeza. *Logical Foundations for Rule-Based Systems*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [7] A. Ligeza, I. Wojnicki, and G. Nalepa. Tab-trees: a case tool for design of extended tabular systems. In H. M. et al., editor, *Database and Expert Systems Applications*, volume 2113 of *LNCS*, pages 422–431. Springer, Berlin, 2001.
- [8] A. Lindsay, D. Dawns, and K. Lunn. Business processes - attempts to find a definition. *Information and Software Technology*, 45(15):1015–1019, December 2003. Elsevier.
- [9] S. Lukichev and G. Wagner. Visual rules modeling. In *Sixth International Andrei Ershov Memorial Conference Perspectives of System Informatics, Novosibirsk, Russia, June 2006*, *LNCS*. Springer, 2005.
- [10] T. McDaniel. Ten pillars of business process management. *eAI Journal*, pages 30–34, November 2001.
- [11] G. J. Nalepa. Business rules design and refinement using the XTT approach. In D. C. Wilson and G. C. J. Sutcliffe, editors, *LAIRS-20: proceedings of the twentieth international Florida Artificial Intelligence Research Society conference*, *FLAIRS*. - Menlo Park, 2007. AAAI Press.
- [12] G. J. Nalepa and A. Ligeza. Conceptual modelling and automated implementation of rule-based systems. In T. S. Krzysztof Zieliński, editor, *Software engineering : evolution and emerging technologies*, volume 130 of *Frontiers in Artificial Intelligence and Applications*, pages 330–340, Amsterdam, 2005. IOS Press.
- [13] G. J. Nalepa and A. Ligeza. A graphical tabular model for rule-based logic programming and verification. *Systems Science*, 31(2):89–95, 2005.
- [14] G. J. Nalepa and A. Ligeza. Prolog-based analysis of tabular rule-based systems with the XTT approach. In G. C. J. Sutcliffe and R. G. Goebel, editors, *FLAIRS 2006 : proceedings of the nineteenth international Florida Artificial Intelligence Research Society conference*, pages 426–431, *FLAIRS*. - Menlo Park, 2006. Florida Artificial Intelligence Research Society, AAAI Press.
- [15] G. J. Nalepa and I. Wojnicki. Visual software modelling with extended rule-based model. In L. A. M. Cesar Gonzalez-Perez, editor, *ENASE 2007: International Working Conference on Evaluation of Novel Approaches to Software Engineering*. INSTICC Press, 2007.
- [16] OMG. Business process modeling notation (bpmn) specification. Technical Report dtc/06-02-01, Object Management Group, February 2006.
- [17] OMG. Semantics of business vocabulary and business rules (sbvr). Technical Report dtc/06-03-02, Object Management Group, 2006.
- [18] R. G. Ross. *Principles of the Business Rule Approach*. Addison-Wesley Professional, 1 edition, 2003.
- [19] 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*, *LNCS*, pages 1–12. Springer, 2003.
- [20] B. von Halle. *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. Wiley, 2001.