

Business rules repository for business rules represented using UML

Olegas Vasilecas, Evaldas Lebedys

Abstract: *The paper shortly discusses the main principles of business rules approach. It briefly analyses business rules modelling techniques. The paper shows the relationship of business rules model with other business models and the importance of business rules model. The paper also discusses how business rules model can be created without externalising business rules from different business models. A part of a business rules repository used to store information about business rules represented in UML Use Case diagrams is presented in the paper.*

Key words: *Business rules, business systems model, business rules modelling, business rules formal representation, business rules repository.*

INTRODUCTION

Business rules approach has become the object of many researches in recent years. As business rules affect different business systems aspects they appear in various systems models. Business rules are hardly externalised from different systems models to separate business rules model as they are integral parts of systems models. Different modelling languages are used to represent different aspects of systems being modelled. Each modelling language is better suited for some types of business rules modelling. It is desirable to have a single coherent representation for all the kinds of business rules, but it is difficult to achieve because business rules differ much in their structure, semantics and other points of view. It is also desirable to represent business rules in a way suitable to perform business rules system consistency analysis. There is no single modelling language suitable to represent business rules of different classes. Even if some modelling language can be used to represent all business rules, some business rule expressions might become hardly understandable. Business rules approach focuses on information systems development automation. Business rules have to be expressed in a way suitable for automated implementation in information systems. On the other hand, business rules should be expressed in natural language sentences for business audience.

We state that it is not necessary to externalise business rules from business systems models to create a detail and full business rules model. We propose to use system models represented using UML diagrams for business rule model creation. Business rules expressed using first order predicate logics are suitable to perform business rules system consistency analysis. In this paper we shortly present the way of business rules deriving from UML Use Case model representing actors obligations. The schema of a part of business rules repository for storing information about business rules that declare actors obligations is also present in the paper.

The rest of the paper is organised as follows. Section 2 surveys the main aspects of business rules approach and presents how business rules appear in different system models. Section 3 shortly analyses business rules representation techniques. Section 4 briefly represents the schema of a part of business rules repository for storing information about business rules expressed in UML Use Case model. Section 5 concludes the paper and presents further work directions.

BUSINESS RULES IN BUSINESS MODEL

Systems analysts have long been describing an enterprise in terms of the structure of the data and the organization of functions organisation performs. But the constraints under which the enterprises operate were ignored in most cases [2]. Semantics captured by

constraints can be expressed as logic rules [6]. There were many concepts of business rule proposed in last few years. In this paper we use the concept of business rule as stated in [8]: business rule means a statement that defines or constrains some aspect of the business. Concept "business" in this context is used as an abstraction that refers to any subject of any scope to which model driven architecture or UML modelling is or could be applied [8]. This means that concept business in this paper is used to specify any type of activity having its goals, using its resources, processing some processes and achieving some results.

Business rules express a crisp logical definition of some facet of the organization's way of doing business [9]. Business rules that drive an organization are almost always expressed as logical knowledge [7]. Business rules are simple atomic statements that constrain some aspect of business operations [9]. Business rules can appear in many forms. They may be described both formally and informally [2]. As mentioned above, business rules express constraints under which business is done, so business rules statements have to be well-formed expressions and not Ad Hoc [5]:

- Business rules statements should be suitable for validation of correctness by business people;
- Business rules statements should be suitable for verification against each other for consistency.

BUSINESS RULES REPRESENTATION

There is no generally accepted approach for defining or representing business rules [9]. Different modelling languages can be used to represent business rules: UML, structured English sentences, and program code [8]. Business rules differ in their structure, semantics, systems components which they are related to. There are many different classes of business rules [1], [4]. Business rules of different classes can be represented differently [10]. It may be hard or even impossible to represent all business rules existing in particular system using one modelling language. Selected modelling language impacts business rules models clarity and suitability to perform verification and consistency analysis. Concept "model" is used here as a representation of a part of the function, structure and/or behaviour of system of interest [9]. A model is generally regarded as a representation of reality [13]. The system in this context may be one of the following: business system, information system, business rule system and so on. Each model uses notation suitable to its primary audience. The syntax and style of a model is dictated by the language, customs, tools, and training of its primary audience [8].

Business rules appear in different system models. Business rules are represented differently in different models. A full business rule model should be complete, correct and consistent. Business rule model should also be clear, unambiguous and readily translatable into other representations. There are formal, semi-formal and informal models. Model is formal if it is represented using a language having well-defined form ("syntax"), meaning ("semantics"), and possibly rules of analysis, inference, or proof for its constructs [9]. Only formal models are of interest for us because only formal models are suitable for automated processing and transformation into other models. Formal logics, such as predicate logic, are suitable to create well-formed expressions of rules in business terms. Rule model created using formal logics is suitable for automated business rule implementation [5]. Formal model can be created using graphical or textual notation [9].

An additional need for the transfer of business rules between organizations also shows the importance of business rules approach [9]. Although all organizations differ they also have some similarities. For example, business rules implementing constraints defined

by different laws appear in every organisation. Business rules transition between organisations may become asset in the future.

UML is the most popular modelling language used to model business systems, information systems and software systems [11], [12]. Business system models represented using UML diagrams are not fully formal. Systems models can be differently interpreted because some details cannot be represented in UML diagrams. Only diagrams are not enough to create full and detail system model. Some aspects of a system are hardly represented using graphical notation or a hardly understandable if represented in this way. The need of a formal technique in addition to graphical notations for modelling different kinds of systems was shown by introducing object constraint language (OCL) in addition to UML. OCL is used in parallel with UML diagrams to model different kinds of systems. Complex logic of the systems being modelled can be described using OCL, but there are no means to check the consistency of the system models and OCL specification together.

REPOSITORY FOR BUSINESS RULES REPRESENTED IN USE CASE MODEL

Many rules existing in various systems are specified representing different aspects of the system: system behaviour, system structure, system actors obligations and so on. But not all business rules can be represented in system models describing different aspects of a system. Some rules have to be additionally specified to create full and detail business rules model.

Business rules influence almost all aspects of business systems. Business rules state, define and constrain structural elements of business systems and behaviour of system components. This is the reason why it is difficult to externalise business rules from different system models [3], [4]. By saying "externalise business rules" we mean the action of moving all business rules from the models representing different aspects of the system to separate business rule model. Some logical parts of the system models are transferred to business rule model when moving business rules to a separate model. This may cause the incompleteness of some business system models.

The other way of creating complete business rule model is duplicating business rules represented in different system models. In this case each business rule is represented twice. Each business rule in different system models would have a copy in business rule model. It is also a shady solution because with the duplication of information arises many additional problems.

We propose not to externalise business rules from the models describing different aspects of business system. Business rules model can be created transforming business rules represented in different system models into formal business rule model. Business rules that are not represented in any system model have to be specified separately (Figure 1). Business system model analysis is performed after all system models are created. Business rules expressed in different system models are represented formally in business rules model. The relationship between business rule statements in rule model and system models is ensured by storing information on every rule in business rule repository. Business rules model and other system models can be used to perform consistency analysis. Complete business system model includes models describing all necessary for certain purpose system aspects business rule model.

The process of business rule transformation and formal business rule specification is shown in Figure 2. Use Case model can be created using any commercial modelling tool allowing storing information in XML file. The information about Use Case model is retrieved from XML file for analysis.

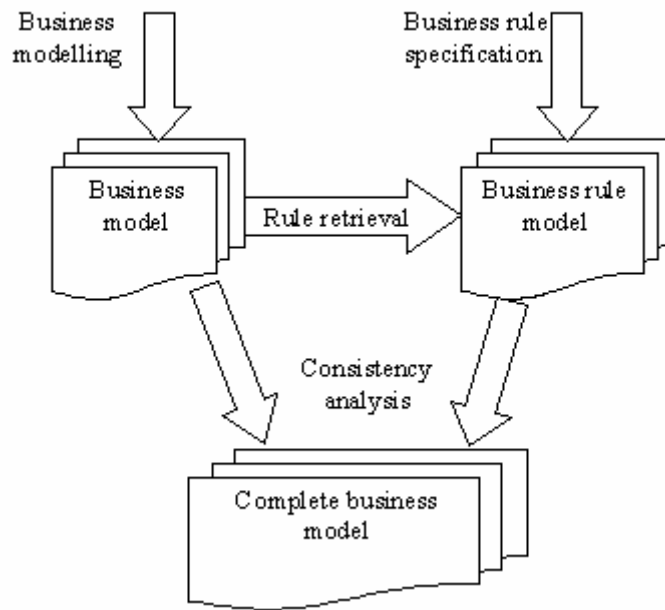


Figure 1. Business system modelling

Information about business rules is stored in rule repository after a search of business rule statements is performed. Formal business rules statements are generated using information stored in business rules repository. Formal business rules statements are also stored in rule repository. Business rules are expressed in natural language sentences. These informal business rules statements can be presented to business audience for business rules verification.

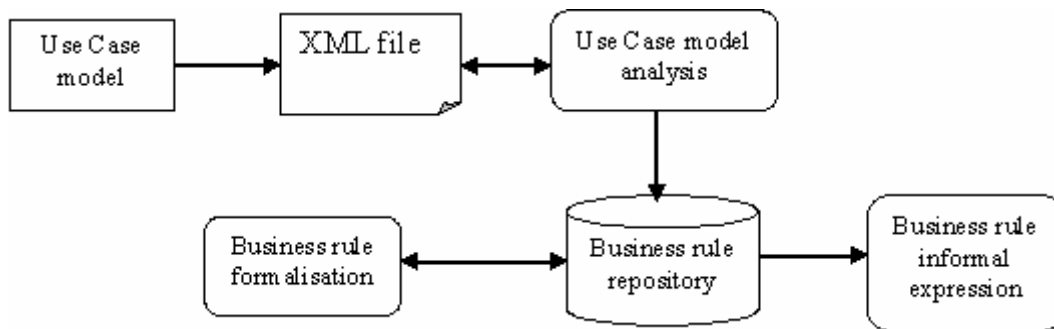


Figure 2. Business rule transformation process

Figure 3 shows the schema of the business rule repository. As we discuss only business rules represented in UML Use Case diagrams in this document, only a part of repository for storing business rules describing actors obligations is represented. We will briefly describe the repository further.

Table "Actors" is used for storing information about business actors specified in Use Case diagrams. Table "Use_Cases" is used for storing information about tasks assigned to business actors which are represented in Use Case diagrams. Table "Obligations" store information about the obligations assigned to business actors. Obligations are specified as associations between business actors and Use Cases. Obligations may be "has to", "may", "can". These tables are filled up when analysing XML file which stores Use case model. The name of each object and original identifiers are transferred into repository from

XML file. Modelling tool assigns original identifiers to each component of the Use Case model. These identifiers are used to ensure the traceability between Use Case model and business rules stored in rule repository.

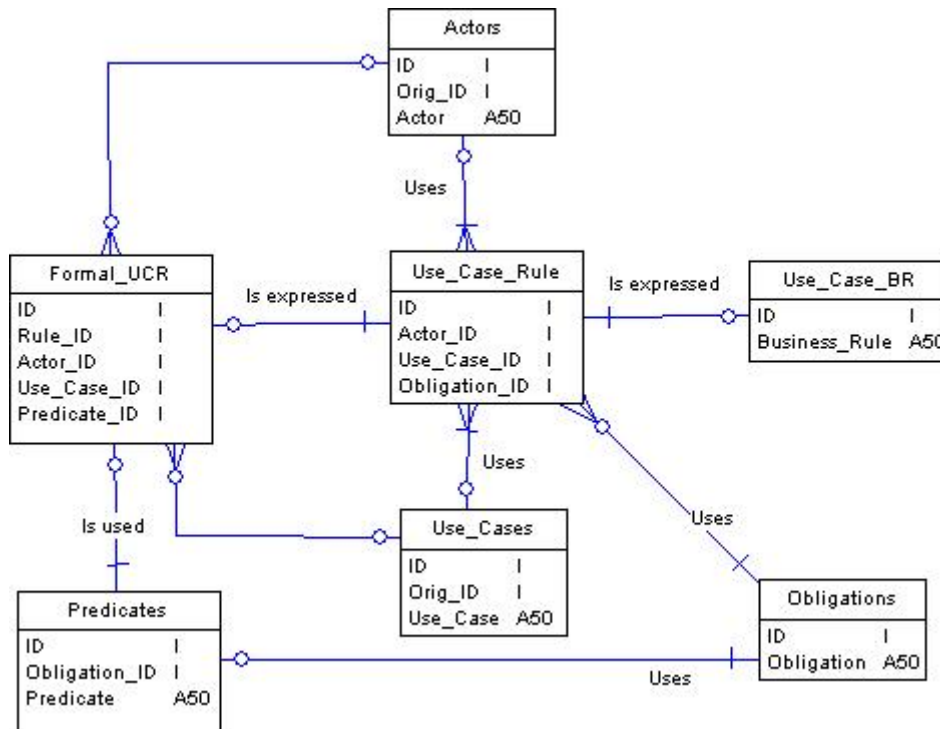


Figure 3. The schema of business rule repository

After the “Actors”, “Use_Cases” and “Obligations” tables are filled up associations between actors and Use Cases are transformed into rules and placed in rule repository. Information stored in table “Use_Case_Rule” is used to form business rules expressions in natural language sentences. These statements are stored in table “Use_Case_BR”. Business rules expressed using natural language sentences are used for rule verification. Business rule verification is performed by business people.

The table “Predicates” is used for storing information about the predicates formally expressing business rules. Predicates are formulated from statements expressing obligations. For example, predicate Has_to(A,U) is formulated from obligation “Has to” and means “Actor A has to perform task U”.

The table “Formal_UCR” is used to store information about formal business rule statements. Formal business rule statements are formulated using predicates stored in table “Predicates”.

CONCLUSIONS AND FUTURE WORK

The importance of business rule approach is obvious. Literature analysis showed that many business rule representation techniques exist but there is no single suitable to model all business rules. The need for different representations of business rules is stated in papers of different authors. Analysis showed that business rules influence almost all aspects of business systems and appear in different business system models. Concluding above presented we proposed not to externalise business rules from different system models. We showed how business rules specifying actors obligations and represented in UML Use Case diagrams can be transferred to business rule repository. We also

presented a part of the business rule repository used to store rules represented in UML Use Case diagrams.

The prototype of the software system implementing proposed above is being developed. Business rule repository is going to be expanded to support storing information about rules represented in other system models. The tool should also provide facilities to perform business rules system consistency analysis.

REFERENCES

- [1] Amghar, Y., M. Meziane, A. Flory. Using Business Rules within a Design Process of Active Databases. *International Journal of Database Management*, 11(3), 2000, p. 3-5.
- [2] Business Rules Group. *Defining Business Rules ~What Are They Really?*, 1995. URL: http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf.
- [3] Cibrán, M. A., M. D'Hondt and others. JAsCo for Linking Business Rules to Object-Oriented Software. *International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications*, Rio de Janeiro, Brazil, June 2003.
- [4] Cibrán, M. A., M. D'Hondt, and others. Aspect-oriented programming for connecting business rules. In *Proceedings of the 6th International Conference on Business Information Systems*, 2003.
- [5] Business Rules Group. *Business rules manifesto*, 2003. URL: <http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf>.
- [6] Liu, Y., C. Pluempitiwiriyaewej and others. A Rule Warehouse System for Knowledge Sharing and Business Collaboration. Technical Report, TR01-006, University of Florida, 2001, URL <http://www.cise.ufl.edu/tech-reports/tech-reports/tr01-abstracts.shtml>.
- [7] Merritt, D. *Best Practices for Rule-Based Application Development*, 2004. URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnma/html/aj1rules.asp>.
- [8] Object Management Group. *Business Rules in Models RFI*, 2002. URL: <http://www.omg.org/docs/ad/02-09-13.pdf>.
- [9] Object Management Group. *Business Semantics of Business Rules RFP*, 2003. URL: <http://www.omg.org/docs/br/03-06-03.pdf>.
- [10] Orriens, B., J. Yang, M. P. Papazoglou. A Framework for Business Rule Driven Service Composition. In *4th Int. Workshop, TES 2003*, B. Benatallah and M.-C. Shan, Eds.: Springer, 2003, p. 14-27.
- [11] Shen, W., K. Compton, J. K. Huggins. A Toolset for Supporting UML Static and Dynamic Model Checking. In *proceedings of the 26th International Computer Software and Applications Conference 2002, Prolonging Software Life: Development and Redevelopment*, Oxford, England, IEEE Computer Society, 2002, p. 147-152.
- [12] Sparks, G. An introduction to modelling software systems using the Unified Modelling Language: The Dynamic Model. URL: http://www.sparxsystems.com.au/WhitePapers/The_Dynamic_Model.pdf.
- [13] Whitman, L., B. Huff. *Issues Encountered Between Model Views. Flexible Automation and Intelligent Manufacturing 1998*. Portland, OR, 1998. p. 117-130.

ABOUT THE AUTHORS

Prof. Olegas Vasilecas, Information Systems Laboratory, Vilnius Gediminas Technical University, Phone: +370 52 744 860, E-mail: olegas@fm.vtu.lt.

Ph. D. Student Evaldas Lebedys, Information Systems Laboratory, Vilnius Gediminas Technical University, Phone: +370 52 744 860, E-mail: evaldas@isl.vtu.lt.