Integration of Business Rules and Model Driven Development

Lauma Jokste

Information Technology Institute, Riga Technical University, Kalku 1, Riga, Latvia
lauma.jokste@rtu.lv

Abstract. To help bridge the gap between Information system (IS) analysts and stakeholders, a Model-Driven Development (MDD) approach is proposed. A MDD approach uses models as primary development artifacts. Models increase the abstraction level of IS development and help to improve migration between various development phases. A MDD approach provides considerable benefits in the IS development domain, nevertheless this approach contains a variety of difficulties. This paper addresses MDD and specifically one of the models used in this approach – the Business Rules Model. Business rules are usually maintained in a textual form thus complicating their usage in the MDD approach. In this paper a new Business rules metamodeling language towards Business rules adaption for MDD is provided. As an input to MDD, Enterprise Modeling (EM) is used.

Keywords: Model Driven Development, MDD, Business Rules, Enterprise modeling, Enterprise Knowledge Development, EKD.

1 Introduction

One of the basic problems in the Information System (IS) development domain and requirements specification is ambiguity between system analysts and stakeholders [1]. System analysts tend to use models to determine the expected result of system functionality, but these models might never be directly used as a component of development process. Software development processes can effectively be improved by using models not only as visual means but also as a software development component. This leads us to *Model Driven Development* (MDD).

MDD is an approach to software development that refers to the systematic use of models and model transformations in entire software development- and runtime. The main idea of MDD is to automate the process of software development by using model transformations [2].

Organizational processes can be described with Enterprise Models. *Enterprise Modeling* (EM) represents behavior, structure, business goals, processes, concepts, actors and resources of an organization. In a context of software development EM

includes representation of system requirements. In the EM approach all processes and components of organization are represented by using conceptual models [3].

This paper represents a Business rules metamodeling language which improves business rules integration with MDD. For describing the components and organization of an enterprise, the EM method *Enterprise Knowledge Development* (EKD) is chosen. EKD is an approach that describes an enterprise as a network of correlated business processes which collectively realize business goals [4]. EKD is selected as a widely used in both business and public sector. EKD has proved its effectiveness by providing a framework for stating, modeling and reasoning regarding pertinent knowledge in difficult problem situations [5].

The EKD approach includes several sub-models in which each sub-model describes the enterprise from different aspects: Goals model, Business rules model, Concepts model, Business process model, Actors and resource model and Technical component and requirements model [3]. In this paper the main emphasis is put on the *Business rules model* (BRM). Business rules are usually expressed in a natural business-like format and they might be liable to frequent changes, which complicates their usage in software development and maintenance. The business rules are means to which an organization is able to control the business, realize competitive strategies, promote the organization's policy and to comply with legal and other obligations [6]. Business rules describe the policies, laws and regulations of an organization. One of the main conditions for MDD is that all models and their components should ensure interoperability between all models used in MDD.

The objective of this paper is to present a Business rules metamodeling language which can be used to integrate business rules in the MDD approach and to discuss its clarity by giving an example case in which a Business rules metamodel is created. Main emphasis is put on a condition that the business rules expressed by provided metamodeling language should be easy perceivable for business people and usable in MDD.

In the context of this paper the metamodel is assumed to be a model's model that serves for the explanation and definition of relationships between various components of the applied model itself [7].

The research taken in this paper is argumentative. Preliminary validation and demonstration of the Business rules metamodeling language is performed using an example case of a student scholarship's system administration requirements, which are based on regulations of scholarship awards.

The remainder of his paper is structured as follows: a background about the business rules and Business rules model is given in section 2. The proposed graphical metamodeling language is given in section 3. In section 4 an example case of Business rules metamodel is demonstrated. In section 5 concluding remarks are given.

2 Business rules and Business rules model

Organization's business rules are usually expressed and maintained in a natural language format. For example, scholarship regulation can contain a business rule:

Only students who have passed all their exams in the previous session on the first attempt are eligible for the Scholarship.

In order to use business rules in MDD, it is necessary to distinguish concepts, attributes, conditions and actors in the rule. A way how company's business rules are described may depend on many circumstances, for example – different lawyers can draw up the same rules and interpret them differently. In order to prevent such situations, we appoint that every rule should be transformed to a standardized natural language form before it is integrated in MDD. For this purpose business rules notation which is based on formal English – RuleSpeak is chosen. RuleSpeak was first developed in 1990 by Business Rules Solutions (BRS), LLC. It is widely used among business people [8] and even been tried for IS requirements specification [9]. It was decided to choose RuleSpeak as a base for a Business rules metamodeling language, because it has a clear definition and defines well structured business rules sentence forms, which can be adapted to Business rules metamodel and MDD.

RuleSpeak is a set of practical guidelines for expressing business rules in unambiguous and well-structured English, which helps to improve communication about business rules among business people, business analysts and IT professionals [8]. The basic concept of RuleSpeak states that every rule should include one of these two words: "must" and "only". Instead of "must", "may" can be used but only when combined with the word "only". RuleSpeak guidelines defines the best practice for business rules sentences structure, how to avoid a redundancy and express rule clearly and unambiguously interpretable. Detailed RuleSpeak guidelines can be found in [10] and [11].

A business rule which is expressed by using RuleSpeak guidelines would look as follows:

The scholarship may be assigned only if student has passed all his exams in previous session on the first attempt.

In the following Business rules model rules are expressed according to a RuleSpeak specification (see in Figure 2.). The Business rules model is developed by using an example case, the requirements of a student scholarship system administration, which is based on regulations of scholarships awards. According to the EKD method, business rules are motivated by goals, they cause business processes and are based on concepts defined in a concept model [12]. In the Business rules model each rule consists of one sentence. In the Business rules metamodel each rule should be divided into separate objects, which our provided graphical business rules metamodeling language supports.

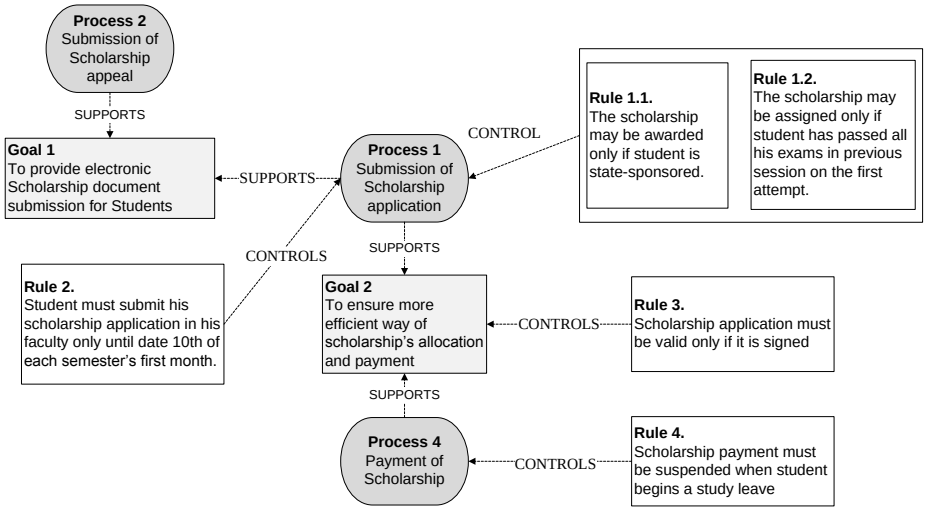


Fig. 1. The Business rules model example case based on the EKD method.

This Business rules model example case is a simplification of real life. There might be more business rules to consider than those displayed in figure 1.

3 Definition of graphical business rules metamodeling language

In this section a graphical metamodeling language is defined. Metamodeling language is based on metamodeling concepts called GOPPRR: graphs, objects, properties, ports, relationships and roles [13]. The Business rules metamodeling language consists of two categories: objects (see Tab. 1) and relationships between them (see Tab. 2).

Objects in a metamodel are connected with links. Every link contains two roles: the source and the target. The source points to an object where the link begins. The target points to an object to which the link is lined. Roles define in which direction the link can be drawn – either directions or only one specific direction. In the provided metamodeling language, roles define from which to which object a link may be drawn. Strictly defined roles and relationships are an important condition for Business rules metamodel to be compatible with other models used in MDD and to be explicitly used in the code generation. Links and roles are defined in Table 2.

Table 1. Defined metamodeling language objects

Object	Description
Rule	A modeling component contains rule name and marks a beginning point of each business rule. Every rule always starts with a *Rule* object. It is necessary for selecting certain rules and for integration with other models used in code generation.

Concept	Concepts define "things" and "phenoma" which are used in all other models [12]. *A Concept* is a modeling component which is used to characterize all concepts used in a Business rules metamodel. Every concept can be defined only once. Concept is, for example: *The Scholarship*.
Action	The RuleSpeak specification says that each business rule must contain one of the words "must" and "may" when describing an action in the business rule, hence the modeling component *Action* is separated in two sections, where the first section contains the *keyword* and the second section *the action* (verb). The *keyword* section contains either predefined values "must" or "may", as well as form of denial: "must not" or "may not". Action is, for example: *may be assigned*, where *may* is keyword and *be assigned* – action.
Additional word	*An additional* word is a word or expression, which helps to create a coherent and readable structure of the business rule. An *Additional word* contains predefined words and word combination lists with values such as *only if, only when, only, if, and, or, in, when, then* etc. *Additional words* usually do not affect business rules modeling functionality but are mostly used to make a logical interpretation of the rule.
Condition	*A Condition* is a composite object, which consists of 4 parts: 1) *Attribute*; 2) *Concept* (*Attribute source*); 3) *Operator*; 4) *Attribute value*. *The Attribute* is a text input field for an attribute name. The *Concept* field must contain one of existing concepts. The *Operator* is a value that can be expressed as a logical operator in code, for example: is equal, is until, is at least etc. The *Attribute value* is a text input field for the value of attribute. Condition is, for example: *Exam attempt is first*, where *Exam* is a concept (attribute source), *attempt* – an attribute, *is* – an operator and *first* – an attribute value.

Table 2. Relationships in graphical Business rules metamodeling language

To: → From: ↓	Rule	Concept	Action	Additional word	Condition
Rule	0	1	0	1	0
Concept	0	0	1	1	0
Action	0	1	0	1	0
Additional word	0	1	0	0	1
Condition	0	0	0	1	0

Relationships between modeling components in a Business rules metamodel are described by a connectivity matrix (See table 2). Value '0' means that a relationship

from one modeling component to other doesn't exist and value '1' means that a relationship between object "From" to object "To" is defined. For example, a link from *Rule* to *Concept* exists, while a link from *Concept* to *Rule* is not possible.

4 Example case

To a purpose to demonstrate provided Business rules metamodeling language, a Business rules metamodel example case is created. Coherence between the Business rules model and its metamodel are shown in figure 2.

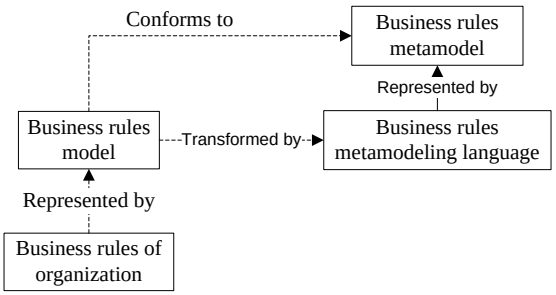


Fig. 2. Coherence between a Business rules model and its metamodel.

In the presented Business rules metamodeling language each rule is designed as an independent structure, where the same objects between different rules are not connected with visible links. Every object or object component should contain a unique identification thus ensuring that, for example, the same named concepts are actually one and the same object. This approach allows building a transparent and easily modifiable metamodel while ensuring metamodel usability in MDD.

Each rule must contain at least one concept, one action and one condition object. Figure 3 shows an example of how to modify business rules expressed in RuleSpeak form to a business rules form in metamodel, using our graphical metamodeling language.

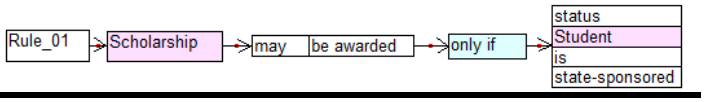
RuleSpeak form	Scholarship may be awarded only if student is state-sponsored.
Form in metamodel	

Fig. 3. Graphical Business rules metamodel representation of example no. 1.

The Business rule expressed in metamodel form contains an extra word *status* (see Fig. 3); it is compulsory that the *condition* object contains an attribute, because it is used to connect the Business rules metamodel with other models. If this rule would be generated back to a formal expression, it would look as follows:

Scholarship may be awarded only if Student status is state-sponsored.

Business rules back-generation from metamodel to their formal expression is a metamodeling tools functionality. It is easy implementable and usable for different metamodeling tools.

The operator field contains the value *is*. For graphical representation, textual operators are recommended, though in code textual operators can easily be defined as logical operators, for example: [is] = [=], [is until] = [≤], [is greater than] = [>], [is at least] = [≥], [is not] = [≠] etc.

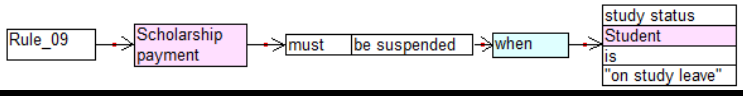
RuleSpeak form	Scholarship payment must be suspended when student begins a study leave.
Form in metamodel	

Fig. 4. Graphical Business rules metamodel representation of example no. 2.

Figure 4 demonstrates a representation of an event-action business rule. If this rule would be generated back to formal expression, it would look as follows:

Scholarship payment must be suspended when Student study status is "on study leave"

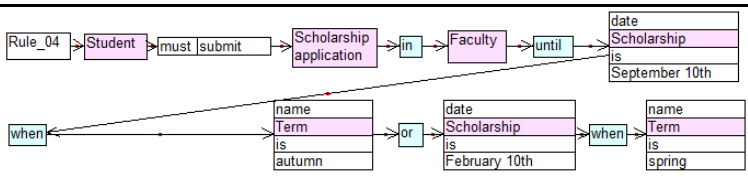
RuleSpeak form	Student must submit the scholarship application to faculty only until the 10th of each term's first month.
Form in metamodel	

Fig. 5. graphical Business rules metamodel representation of example no. 3.

Figure 5 demonstrates a representation of a derivative business rule. A phrase "*until the 10th of each term's first month*" might be ambiguous for developers, therefore to make it unmistakably clear, the number of terms is specified with *each term's first month*. Consequently, if this rule is generated back to a formal expression, it looks like this:

Student must submit scholarship application in Faculty until Scholarship application dates is September 10th when Term name is autumn or Scholarship application date is February 10th when Term name is spring.

In both examples that figures 4 and 5 demonstrate, the business rule sentence form has changed, preserving an essence of the business rule in its RuleSpeak sentence format.

Business rules metamodeling language can be used in different metamodeling platforms, for example Eclipse Modeling Framework (EMF) [14], ADOxx [15], MetaEdit+ [13] and others. It was decided to choose MetaEdit+ as it has strengths in metamodeling and it provides a full functionality for defining a graphical metamodeling language [16].

A fragment of the developed Business rules metamodel in MetaEdit+ is shown in Figure 6.

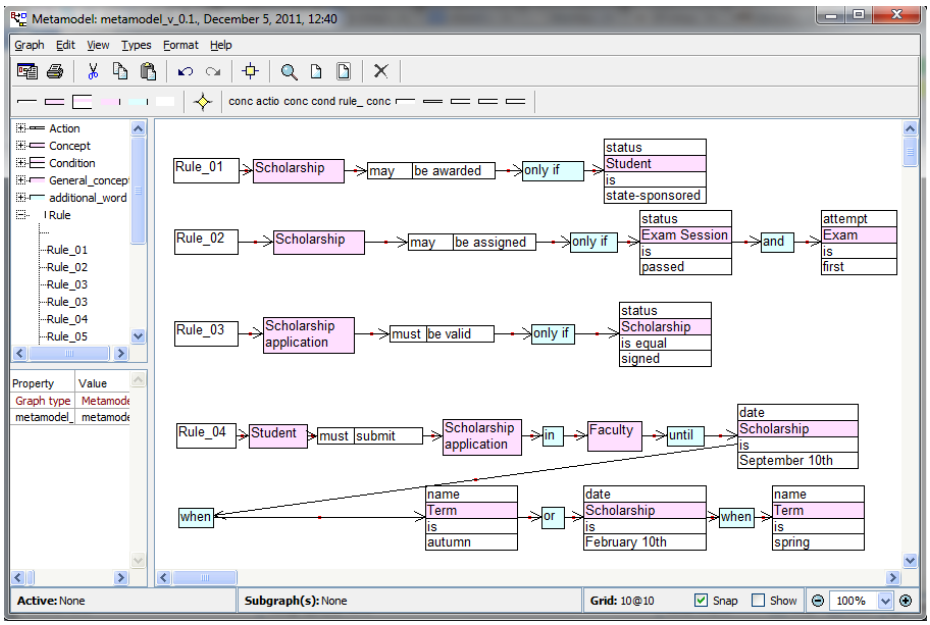


Fig. 6. Fragment of the Business rules metamodel in MetaEdit+ .

5 Concluding remarks

This paper presents a business rules metamodeling language which can be used to create Business rules metamodels for MDD. The most important challenges for business rules usage in MDD is the fact that business rules are usually expressed in a business-friendly manner and are not guided to specific instructions how the business rule should be written. The goal is to clarify the logic of a business rule and to transform it to more rigorous form thus making business rule usable in MDD.

Provided graphical business rules metamodeling language has some similarities with Decision Model developed by Halle von Barbara. Both approaches ensure well-formed, predictable, stable and maintainable expression of the business rules [17]. Unlike Decision Model, Business rules metamodeling language is tended to a logic of the separate objects in the business rule, while Decision Model are based on a business logic in general. But both – Decision Model and Business rules metamodeling language can be anchored to any other models and in the same time can be maintained independently of them.

The provided Business rules metamodeling language is experimentally validated with an example case and ensures that business rules can be transformed from their natural expression which is formalized by RuleSpeak to the Business rules metamodel and vice versa without losing interpretation of the business rule, even if the structure of the rule is slightly changed. For the time being, interpretation changes can be

evaluated subjectively and there is no appropriate statistical method for measuring an interpretation of textually expressed business rules. Business rules expressed by provided Business rules metamodeling language can be further used for code generation in MDD.

However, a more detailed case study is necessary to improve that Business rules metamodeling language can ensure that any kind of business rules can be expressed by this language.

References

1. Kirikova M., Bubenko J.A. (1994), Software Requirements Acquisition through Enterprise Modeling. Paper presented at 6th Int. Conference on Software Engeneering – SEKE'94, Jurmala, Latvia.
2. Sanchez P., Ana Moreira A., Fuentes L. et al (2010) Model-driven development for early aspects, Information and Software Technology 52 p.249–273.
3. Zikra I., Stirna J., Zdravkovic J. (2011.), Analyzing the Integration between Requirements and Models in Model Driven Development, Springer Verlag, Berlin pp 342 – 356.
4. Kavakli V., Loucopoulos P. (1999), Goal-Driven Business Process Analysis Application in Electricity Deregulation. Information Systems 24, pp 187-207.
5. Gints Stale, Ivars Majors (2012), The Application of EM for Knowledge Flow Analyses and the Development of an Educational IT Ecosystem, Vol. 933. Paper presented at the 5th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling, Rostock, Germany, 7-8 Nov 2012.
6. von Halle B., Goldberg L., Zachman J. (2006), Business Rule Revalution. Running Business the right way. Happy about, California - P. 221.
7. Génov G. (2009), What is a metamodel: the OMG's metamodeling infrastructure // Modeling and metamodeling in Model Driven Development, Poland, Warsaw, 14-15 May 2009.
8. RuleSpeak official homepage http://www.rulespeak.com/en/ Accessed 26 Jun 2013.
9. Kapočius K., Danikauskas T. (2006), The Use of Business Rules for the Specification of Dynamic Aspects of IS. ISSN 1392 – 124X, Information Technology and Control, Vol.35., No 3A, pp 327-332.
10. Ronald G. Ross, Basic RuleSpeak Guidelines, Business Rules Solutions, LLC, version 2.2. (2001.-2009.).
11. Ross R. G. (2001.-2009.), RuleSpeak Sentence Forms, Business Rules Solutions, LLC, version 2.2.
12. Bubenko, J., Persson, A., Stirna, J. (1996), User Guide of the Knowledge Management Approach Using Enterprise Knowledge Patterns, no IST 2000-28401, Sweden, ftp://ftp.dsv.su.se/users/js/ekd_user_guide.pdf Accessed 10 Jun 2013.
13. MetaEdit+ Domain-Specific Modeling environment // Metacase (2012). http://www.metacase.com/MetaEdit.html last accessed 10.06.2013.

14. Eclipse Modeling Framework Project, http://www.eclipse.org/modeling/emf/ Accessed 27 Aug 2013.
15. ADOxx, http://www.adoxx.org/live/home Accessed 27 Aug 2013.
16. Kern H. (2008) The Interchange of (Meta)Models between MetaEdit+ and Eclipse EMF Using M3-Level-Based Bridges. Paper presented at the 8th OOPSLA Workshop on Domain-Specific Modeling conference. Nashville, TN, USA, 19-20 October 2008
17. von Halle, B., Goldberg, L. (2009) The Decision Model: A Business Logic Framework Linking Business and Technology. Auerbach Publications.