Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 949–952
ISBN 978-83-60810-51-4

Model Driven Architecture and classification of business rules modelling languages

Bartłomiej Gaweł
AGH University of Science and Technology
in Krakow
ul.Gramatyka 10, 30-067 Krakow, Poland
Email: bgawel@zarz.agh.edu.pl

Iwona Skalna
AGH University of Science and Technology
in Krakow
ul.Gramatyka 10, 30-067 Krakow, Poland
Email: skalna@zarz.agh.edu.pl

***Abstract*—An organisation's activity under dynamic changes of business processes requires continuous improvement of business practices. This implies the necessity of refining decision making process. Business rules [6], [8] enable experts to transfer enterprise strategy onto the operational level using simple sentences which, in turn, can automate reactions to subsequent events both inside and outside the organisation. The main advantage of the business rules is their simplicity and flexibility so they can be easily utilised by different organisations for different purposes. In order to represent knowledge in a pseudo-natural language understandable to information systems (business rules engines) notation and description standards are required. In this study, an overview and classification of the most popular business rules description languages are presented.**

I. Introduction

THE central idea behind Business Rules Engines (BRE) or Business Rules Management Systems (BRMS) is that any organisation uses a decision logic to maintain operational and managerial tasks. The major objective of modern business rules management systems is to allow for business and technological processes to be described independently of the software system to be implemented. Nowadays, it is possible to automate business processes by implementing appropriate ERP, CRM, SCM or WFM solution. However, there are still some processes and decisions which are non-routine or cannot be well described by appropriate semantic language.

Rules play an important role among the most frequently used knowledge representation techniques. They may be used to describe different aspects of business. The biggest advantage of BRMS approach is that business rules are understandable for human and may be also directly used in software programs [4]. However, software programs and humans need rules to be expressed in different languages in order to understand them.

Different target audience and broad field of potential applications cause that there are a number of languages and approaches for modelling of business rules. This article presents some criteria which may be used to select business rules modelling language being the most suitable for specific problem.

The paper has the following structure. In Section II, preliminary information on business rules and Semantic Webs are described. Classification of business rule description languages is discussed in Sections III and IV. The paper ends with concluding remarks.

II. Business rules and semantics web

The definition of a business rule, coming from GUIDE [11], states that 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 behaviour of the business*." In [10], Ross describes several basic principles of business rules approach. He believes that a language has the biggest impact on business rules expressiveness. Therefore, in the remaining part of this study business rules description languages are presented and classified.

Business rules are usually written in a form of "*if... then ...*" statement. The most important aspects of business rules is that they should be unambiguous. It means that all terms used in rules should be defined in a business vocabulary which defines all important concepts in a particular area of business interest. Besides the definitions of concepts, such vocabulary also defines relations between concepts. A formal description of rules and a vocabulary should be different for different target users.

From managerial point of view, a goal of business rules approach is to improve the communication between humans. Hence, business rules and business vocabularies should be described in a language close to the natural one. Business rules management systems need a definition of a vocabulary and rules in formal language to avoid ambiguity of meaning. Nowadays, especially in supply chains, business rules managements systems participate in larger networks and exchange business rules. In such a case, an information context should be transferred by a language together with rules and vocabulary. Unfortunately, there are no languages which meets all those requirements. To solve that problem, two models - *Model Driven Architecture*[1] (MDA) and *Semantics Models* (SM) have been developed.

MDA model defines three abstraction levels. *Computation Independent Model* (CIM) shows how the system works within an environment, without details on the system's structure and application implementation. At this level, rules are written in a language with the syntax close to the natural one or are presented as diagrams (visual languages). *Platform In-*

[1]A detailed description of the MDA architecture can be found at http://www.omg.org/mda/

dependent Model (PIM) is a model of a software system or a business system that is independent of the specific technological platform used to implement it. At this level, rules have standard representation based on an XML language. Finally, at the *Platform Specific Model* (PSM) rules are described in a language designed for a specific BRMS. The idea of SM focuses on exchanging data between systems without human support. Hence, Semantic Models develop languages which transfer both data and context between those systems.

MDA and SM concepts assume that there should be many very limited purpose languages and the appropriate translation programmes. The next section describes the differences between rules description languages.

III. Classification of business rule modelling languages

On the basis of the subject literature [1], [2], [3], [9], [12], the following set of the most popular modelling languages have been selected for the comparison purposes: the Semantics of Business Vocabulary and Business Rules (SBVR), the REWERSE Rule Markup Language (R2ML), the Semantic Web Rules Language (SWRL), the Rule Markup Language (RuleML), the Rule Interchange Format (RIF), the Java Specification Request (JSR-94), Prova and the Production Rule Representation (PRR).

First, the comparison of business rules description languages is performed at the respective abstraction level of the MDA architecture and the selected languages are assigned to those levels:

- **Business models level CIM**: SVBR language,
- **Business functionality level PIM**: PRR, R2ML, SWRL, RuleML, RIF languages,
- **PSM level**: JSR-94 language.

One can see that SVBR is the only standard that exists at the CIM level. It enables rules to be described in a language similar to a natural language.

At the PIM level there are five languages with different expression power. Thus, at the PIM level there is no one universal language which means that the language must be selected appropriately to specific rule type and vocabulary. RuleML is the only language that supports modelling of any rule. Strict assignment of languages to an appropriate type of rule is presented later in this section. Here, it should be underlined that all business rules description languages at the PIM level are written down using XML.

Languages at the PIM level are then transformed into languages at the PSM level. At this level rules modelling languages are strictly connected with programming languages used for developing specific BRMS. This is due to the fact that at this level, rules should be written in a way which guarantees computational effectiveness. Most of the BRMS (e.g., Drools, ILog) is written in Java. Hence, JSR-94 is a standard at the PSM level as an API Java library.

Next, a comparison of meta-languages used to write down standards of business rule modelling were made. The SBVR model is written using a natural (English) language, and dependencies between themes are written using the XML and SBVR XML Schema. Two meta-languages can be distinguished at the PIM level. Rules are written in XML, and respective description schemes in MOF/UML (PRR, R2ML, RuleML), or EBNF formalism (RuleML, RIF, SWRL). At the PSM level, languages are described in a way which guarantees their compatibility with the JSR-94 standard or Java.

Another comparison checks how business rules modelling languages *integrate with external ontologies or databases*. The integrations are usually implemented through establishing links to dictionaries contained in ontologies. These links are most often implemented through URI or IRI. Common usage of URL (*Uniform Resource Locator*) in a semantic model dates back to the work by Heflin and Hendler [7].

A dictionary is seen as an external data type and is represented by external objects. Owing to this solution, rules description languages retain quite a simple construction. In most of the markup based languages (R2ML, RuleML, Jena Rules, OCL, RIF, SWRL), URI identifiers are used to identify facts. Some of them, such as SWRL, have special markups (e.g., `swrlb:resolveURI`) which enable operations on arguments described by URI. All classes described by R2ML have also URI references. Class is a type for R2ML objects or variables. Similarly, reference property and datatype predicate in R2ML have URI references. RIF uses IRI as constants, similar to RDF.

The comparison described above clearly shows that there is one standard, SBVR, which allows the production of business vocabularies and rules to be written in a language close to the natural one. This standard is dedicated to human audience. The same role for computer audience plays JSR-94 closely related with Java. Both of these languages may be easily translated to PIM level languages. Because of different expressiveness of PIM level languages, the choice of the inappropriate intermediate translation language may result in incomplete translation or translation lost. Therefore, the following part of the paper focuses on the following languages from the PIM level: R2ML, SWRL, RuleML, RIF.

IV. Comparison of structures of PIM level modelling languages

A knowledge database is a structure which consists of several levels. For the PIM level languages comparison purposes, the three among the latter are the most important: data level, rules level and knowledge representation level. At the data level, *methods for building particular languages* should be compared. Scheme type and validation are reliable comparison criteria.

Dependencies between structures in business rules modelling languages are usually described using XML Scheme. There are two approaches to build the latter. A language can be described by a flat scheme or by a structure in which each construct is described by a separate module, which is then used to build ontology. The second hierarchical approach, unlike flat structures, can easily be extended. Moreover:

- a programmer needs to use necessary modules only (e.g. those adjusted to some type of rules), he does not have to use all elements,
- it is much easier to develop a language described by a modular scheme,
- modular approach facilitates testing of the solution used to implement the model.

R2ML are based on flat scheme. On the other hand RuleML, SWRL, and RIF follow the second approach. They define respective constructs in separate modules.

Rules are built on facts and terms which creates business vocabulary. A business term is a word or phrase that has a specific meaning for a business in some designated context. A fact asserts an association between two or more terms. That is, it expresses a relationship between the terms. Business rules description languages on PIM level enable a flexible facts notation. Fact can be represented by: term, class, object instance, state or event. A fact can be defined directly in a language or can be assigned to a rule through a proper addressing language URI and IRI (described above).

One of the most important question concerning directly defined facts is the question of whether facts and terms are stored as attributes or elements. If business rule contains information which can be validated, then an element should be used. Otherwise, one can use attributes.

R2ML differs from other languages at the PIM level, because it writes all information items as attributes in contrary to other languages which uses elements as well. However, all of the PIM level languages enable the facts to be described as simple and complex datatypes. For example, beside simple representation of an attribute, RuleML introduces possibility of organising attributes in multisets (`arg`) and sequences (`slot`) using the index attribute. Similar notation is used by SWRL and RIF. R2ML does not distinguish between attribute and element notation. Direct defined facts are described in different ways in different languages. RuleML provides a certain set of constructs which can be used in any semantically significant combination. Thanks to this, RuleML rules are easy to write and maintain, and can be automatically translated into other languages. SWRL and RIF, which are based on R2ML, introduce more complex structures. In contrary to RuleML, R2ML very strongly differentiates a type of terms and atoms. Owing to this, R2ML has a very rich and extensive syntax. For example, a variable in R2ML can be represented by an variable responsible for objects, literals, and (variable without a type). In RuleML and its derivatives, there are no such division. To distinguish between an object and a value, R2ML, similarly to RuleML, introduces respective objects for an object name and a value of a variable.

Languages for writing rules in business rule engines are usually based on a Prolog (Prova, TRIPLE, or Jena) syntax or on their own syntax adjusted to a programming language used to write the engine.

There is a wide range of business rules applications. Therefore, different type of rules can be distinguished [12]:

- *integrity rules* are used to express conditions that should be fulfilled, e.g., "*Each new person driving a staff car must be a qualified driver*".
- *deriviation rules* describe how data item can be computed from other data, e.g., "*A customer is considered a premium customer if it spends more then USD 3,000 a year*".
- *reaction rules* respond to an event occurrence, e.g., "*If a customer has returned a rental car and the mileage is more then 8,000 kilometres since last servicing, then send the car to the service*".
- *production rules*, in contrary to reaction rules, are not called by any previous event, they "produce" events if the conditions are fulfilled. An example of the production rule at the CIM level could be: "If order value is greater than 2,000 and the type of customer is not Premium, then grant a 5% discount".
- *transformation rules* describe constraints which must be fulfilled to change operational data format, e.g, "*If you receive a full description of the book, then transform it to a short description containing the author name and book title only*".

Table I contains an overview of how particular description languages support the execution of different types of rules. Realisation of particular types of rules is possible through the introduction of what is known as procedural attachments and built-in functions. A business rule must often recall queries and procedures stored externally and call external sources of data. Such procedures, called procedural attachments, can be used in business rules.

A procedural attachment is a function or a predicate which is implemented externally. There are two types of such attachments:

- logical attachment, which returns `true` or `false`,
- object, which gets certain objects and returns one or more objects.

In addition to procedural attachments, the built-in function can be mentioned. They are functions built in rule description languages, or predicates which enable operations to be performed on, e.g., strings, numbers or logical values.

SWRL and RuleML have the common library of built-in functions, which is based on the functions available in XQuery and XPath languages. Built-in functions are called in SWRL by a special atom '`swrlx:builtinAtom`'.

SWRL does not support procedural attachments, but they might be called directly from ReactionRuleML. R2ML supports built-in functions by default, in the same way as SWRL does, but uses `r2ml:DatatypePredicateAtom` predicates and `r2ml:DatatypeFunctionTerm` functions. Besides the function, an `r2ml:DatatypePredicateAtom` element contains variable name and type. Thanks to this, in the next step, it can be projected on the respective operator with the respective data.

R2ML enables the introduction of procedural add-ins that, as depicted in the listing, allows access to any external function. To represent production rules, R2ML was enhanced

TABLE I
COMPARISON OF HOW PARTICULAR LANGUAGES SUPPORT OCCURRENCES OF SUBSEQUENT RULES

Language	Rule type				
	Integrity rule	Derivation rule	Reaction rule	Production rule	Transformation rule
R2ML	X	X	X	X	
RuleML	X	X	X	X	X
SQL	X	X	X		
Jena Rules	X	X	X		X
OCL	X	X	X(XOCL)	X(XOCL)	
RIF	X	X			
SWRL	X	X			
PROVA		X	X		

with constructions which enable action-reaction rules to be described.

Knowledge representation is here understood as a way of presenting knowledge about the world together with processing, especially inference, methods. From the business rules' point of view, the following knowledge representations can be distinguished [5]:

- *decision table* - represents knowledge in a form of a table, which contains prerequisites and conclusions. This kind of knowledge description supports the creation of models containing numerous independent conditions,
- *decision grid* - supports the presentation of rules which are functions of two or more related conditions,
- *decision tree* - presents rules as a tree-like graph which shows decision process. In the case of numerous tied rules, a decision tree clearly shows the history of decisions. The main deficiency of decision trees is that they do not support the application of advanced solutions, such as AND or XOR, for controlling processes,
- *scenarios* - describe stereotypical sequences of process events using special scenes corresponding to possible decision situations.
- *workflows* - represents rules as a set of actions that are executed when certain specified conditions are met. Workflows were adopted for presentation of knowledge using tools for description of system workflow, a timing of system events can be presented.

All described standards enable rules to be combined in the form of a decision table. Subsequent rules are written one under another in the form of an XML structure. Rules cannot be written in the form of a workflow in any of the standards. Rules can be written in the form of a workflow using the token language BPEL, which supports combining decision rules.

V. CONCLUSION

The most important business rule description standards are presented in this study. Selected from the recent literature, the most common business rule modelling languages, such as SBVR, R2ML, SWRL, RuleML, RIF, JSR-94, Prova, PPR, have been characterised and compared. In the first step, they were grouped to account for different goals they realise and to make it possible to compare them. The MDA architecture and Semantic Web were selected as classification criteria to create groups of comparable languages. Based on the MDA classification, the following conclusions can be drawn:

- First of all, it should be noticed that there are no useful business rule description standards at the CIM level,
- At the PSM level there exist one standard only for communication with business rules engines - JSR-94. It should be emphasised that it is compatible with the Java language,
- There are no languages supporting efficient business rule visualisation
- There is a lot of languages at the PIM level; however, none of them can be translated to SVBR,
- Most of the standards do not support writing all types of business rules - RuleML is an exception,
- There are no languages based on SQL,
- It is difficult to draw conclusions based on the XML rule notation. The SQL notation would be more accurate.

REFERENCES

[1] G. Antoniou, C. Damasio, B. Grosof, I. Horrocks, M. Kifer, J. Maluszynski, P. Patel-Schneider: "Combining Rules and Ontologies – A Survey." Deliverables I3-D3, REWERSE, http://rewerse.net/deliverables/m12/i3-d3.pdf, March 2005.

[2] H. Boley: "Are Your Rules Online? Four Web Rule Essentials.", RuleML 2007, pp. 7–24.

[3] F. Bry, M. Marchiori: "Ten Theses on Logic Languages for the Semantic Web", pp. 42–49, PPSWR 2005.

[4] E. Gottesdiener, "Business RULES show power, Promise", *in: Issue of Application Development Trends*, vol. 4., no. 3, 1997.

[5] I. Graham, *Business Rules Management And Service Oriented Architecture: A Pattern Language*, Wiley, 2007.

[6] T. Halpin, "Business Rules and Object Role Modeling", it in: issue of Database Programming & Design, vol. 9, no. 10, 1996, pp. 66–72.

[7] J. Heflin, J. Hendler, S. Luke: SHOE: A Knowledge Representation Language for Internet Applications, Technical Report CS-TR-4078, (UMIACS TR-99-71), 1999.

[8] T. Morgan, *Business Rules and Information Systems*, Boston: Addison-Wesley Publishing, 2002.

[9] A. Paschke, *Rule-Based Service Level Agreements – Knowledge Representation for Automated e-Contract, SLA and Policy Management"*, IDEA Verlag GmbH, München, 2006.

[10] R. Ross, *Principles of the Business Rule Approach* , Addison-Wesley Information Technology Series, 2003.

[11] The Business Rules Group, "Defining Business Rules, What are they really?", http://www.businessrulesgroup.org, July 2001.

[12] G. Wagner, C. Damasio, G. Antoniou, "Towards a general web rule language", *International Journal of Web Engineering and Technology archive*, vol. 2, 2005, pp. 181–206.