Visual Rules Modeling

Sergey Lukichev[1], Gerd Wagner[1]

Brandenburg University of Technology at Cottbus
{Lukichev, G.Wagner}@tu-cottbus.de

1 Introduction

Rules are widely recognized to play an important role in the Semantic Web. They are a critical technology component for the early adoption and applications of knowledge-based techniques in e-business, especially enterprize integration and B2B e-commerce. This includes, in particular, markup languages for integrity and derivation rules, such as the Semantic Web Rule Language (SWRL)[5] that has recently been proposed as an extension of the Web ontology language OWL[4]. Rules also play an important role in information systems engineering, especially in the specification of functional requirements where business rules are the foundation for capturing and modeling business application logic.

A lot of work has been conducted in the area of visual representation of business vocabularies. The mainstream technology is MOF[9]/UML[10], which allows visualization of domain concepts by means of, for instance, UML class diagrams.

On the other hand, relatively few research has been done in the area of visual rules modeling. The emerging technologies for the Semantic Web, where rules play an important role, experience lack of modeling tools for visual representation of ontologies and rules. The request for a UML-based rule modeling tool for the Semantic Web comes from the industry. Many companies claim that even if they understand benefits of using Semantic Web technologies like ontologies and rule languages, it is difficult for them to start since ontology architects and rule experts are quite expensive. A UML-based rule modeling approach for the Semantic Web will facilitate the use of the Semantic Web technologies by traditional UML modelers.

The actuality of the proposed research also comes from the rules standardization efforts of W3C (`http://www.w3.org/2005/rules`) and OMG (`http://www.omg.org`), which need rules modeling methodologies and tools.

This paper gives a quick overview of existing rule modeling solutions (Section 3) and presents visual rules modeling approach, based on MOF/UML, using examples (Section 4). In the conclusion part we formulate main advantages of our research in progress (Section 5).

2 Research Description

There is a general problem of interaction between domain experts and technicians, who formalize a business domain and business requirements. To contribute to the solution of this problem we work on methodologies for visual representation of rules, which intend to help capturing business rules from a natural language to the visual/formal representation.

Our present and future work intends to give appropriate answers, for instance, to the following questions:

1. How extensible is UML to support rules in diagrams?
2. How can we integrate the visual modeling of rules with existent modeling tools?
3. Is it possible to adopt component SE and aspect SE to ontology and rules modeling in order to deal with large ontologies and different rule systems? This question address a well-known problem of business rules management and business rules validation in large rule-based systems.
4. What are the relations between UML/OCL and OWL/SWRL? Can OWL and SWRL be transferred into UML/OCL and vise versa in order to exchange rules between two communities of UML modelers and ontology architects?

3 Current Knowledge of the Problem Domain

Existing UML modeling tools usually provide facilities for class and relationship modeling. These models have a static and declarative nature and cannot be used for modeling reactive nature of the Semantic Web in particular and rules-driven business processes in general. The Object Constraint Language (OCL [1])is used for expressing rules in UML class diagrams. Existing tools support serialization of OCL constraints to XMI and there are efforts of Java code generation directly from UML class diagrams with OCL constraints. The latest is supported by Fujaba Tool Suit (`http://www.fujaba.de`).

German company Visual Rules (`http://www.visual-rules.com`) provides a tool for visual modeling of rules in block-schema like style, which may cover some types of business rules.

Market leaders in business rules solutions, ILOG (`http://www.ilog.com`) and LibRT (`http://www.librt.com`), provide flexible tools for rules modeling and deployment, but contain no visual modeling components, which complicates development of rule-based applications.

Concerning Semantic Web technologies, there are several methods for rules modeling.

The Protege tool (`http://protege.stanford.edu/`) provides facilities for ontology and rules modeling. In particular, it supports modeling in RDF and OWL as well as modeling of SWRL rules. In conjunction with reasoning engine, the tool can be used for consistency check of ontologies and serialization to the rule markup. Protege is not a visual tool and requires a significant knowledge of ontology modeling. Moreover it is doubtful that it can be easily adopted in enterprizes, which already use UML technologies for software engineering.

There are ontology language specific tools for visual representation of ontologies, for instance, SemTalk from Semtation GmbH (`http://www.semtation.de`), which provides a visual language for modeling of OWL ontologies. The approach of defining visual language for a particular ontology language has a lack of flexibility and scalability, while our UML-based approach has a power of MDA and allows obtain rules in language-independent manner.

In general, our main activities are focused on development of new visual notations for vocabularies and rules. We consider rules on top of UML class diagrams because they are widely used in software development and such rule modeling principles can be easily adopted by large community of UML modelers.

According to Business Rules Manifesto[6], rules are build on top of vocabularies. This is why extending UML, which is used to express business vocabularies, with a concept of a rule is natural.

4 The State of Art

Main classes of rules at three different abstraction levels are depicted on Figure 1.

More detailed description of rules classification is provided in [7] and defines, in particular, derivation rules, production rules, reaction rules and integrity constraints.

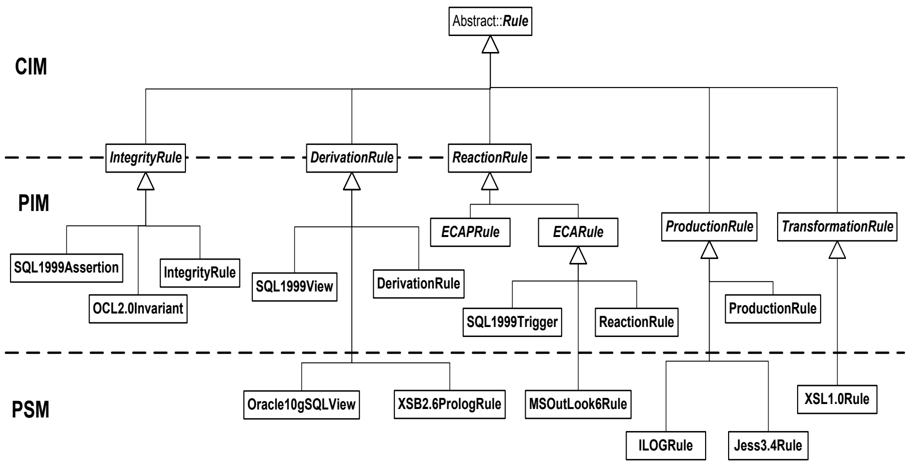

Fig. 1. Rule concepts at three different abstraction levels: computation-independent (CIM), platform-independent (PIM) and platform-specific (PSM) modeling.

In order to support modeling of these rules in UML, a *UML-Based Rule Modeling Language (URML)* [1] has been developed, which extends UML metamodel with a notion of a rule and defines a visual notation for rules.

[1] The URML on I1 website http://www.rewerse.net/I1 or in REWERSE I1 deliverable D8.

In order to exchange rules between communities of UML modelers and ontology architects, the *rule markup framework R2ML* ([8]) has been developed. The R2ML accommodates main concepts of UML/OCL and OWL/SWRL, which allows rules capturing, expressed in datalog-like languages (f.e. SWRL) and functional languages (OCL). The visual rules modeling tool "Strelka" for derivation rules, production rules and reaction rules is currently under development[2]. The tool supports URML as a visual language for rules and serializes rule models into R2ML, which allows rules deployment into rule systems and rule reasoners.

As an example of the visual modeling of derivation rules, let's consider the rule formulation by domain expert in a natural language and rule visualization in a case tool.

Let's consider a rule example from the EU-Rent case study[2]:

> *If return branch of rental is different from pickup branch, then the rental is one way rental.*

This is a *derivation rule*, which specifies how *one way rental* class is derived. The part of a business vocabulary, visualized by means of a class diagram is denoted on Fig. 2.

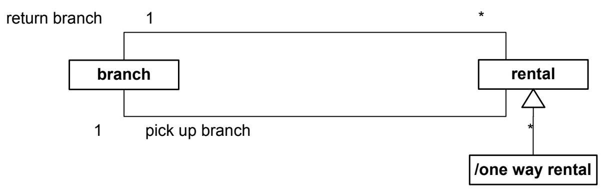

Fig. 2. Part of the EU-Rent business vocabulary

The OCL expression of this rule is:

```
context Rental inv:
if self.returnBranch<>self.pickupBranch
then self.oclIsKindOf(OneWayRental) endif
```

For visual representation of rules we introduce the following URML constructs:

Derivation rule expressed graphically as a circle with a rule identifier,

Condition arrow defines a relationship between a model element that is conditioned and the rule. An example of conditioned model element is a class or an association.

[2] Strelka tool description: http://rewerse.net/events/annual-meeting-2006/demos/i1-demos.html,
Strelka home page on I1 website: http://www.rewerse.net/I1

Conclusion arrow defines a relationship between the rule and a derived model element. An example of derived model element is a class, an association or an attribute.

Using this modeling notation we may visualize the rule as depicted in Fig. 3. The boolean expression *returnBranch* <> *pickupBranch* at the beginning of

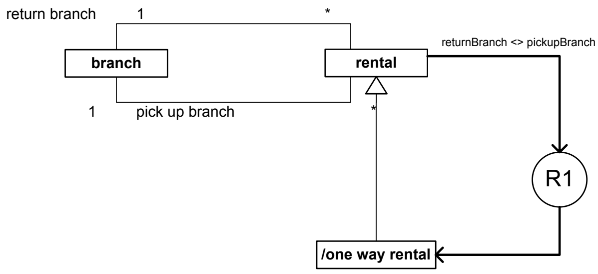

Fig. 3. Visualization of a derivation rule

the condition arrow is a filter. It filters rental objects with different return and pickup branches. Using this approach we may visualize different derivation rules, where classes, associations or attributes are derived. This visual representation corresponds to the following logical formula, where "." is a function, which return attribute value for an object:

$$x \in OneWayRental \longleftarrow x \in Rental \ and \ x.returnBranch <> x.pickupBranch$$

As can be seen from the example, the visualization of the rule is vivid and simple. For detailed description of the rules metamodel and more examples we refer to the website of the REWERSE Working Group I1.

Another important class of rules under consideration is production rules. These rules are widely used in business process automation, supported by several commercial tools and under standardization procedure of W3C.

As an example let's consider the following rule:

> *If the total amount of shopping cart of a customer is more than 100 give customer a voucher with value 10.*

The voucher is created by means of so called *CreateAction* (denoted by character "C" near the arrow head) with a set of initialization parameters (Fig 4). The value 4.8 near the rule circle is a unique rule identifier in a rule set.

5 Conclusion

Main advantages of the introduced rules visualization approach against technologies, described in Section 3, are:

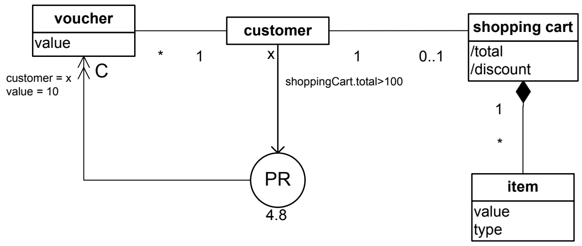

Fig. 4. Visualization of a production rule

Simplicity - with relatively small and simple extension of UML metamodel, visual modeling of main rule types can be implemented.

Visualization - visual representation of rules facilitates the use of rule-based technologies.

W3C Semantic Web and OMG MOF technologies - the solution under development for visual rules modeling may connect widely used OMG MOF methodologies with emerging Semantic Web technologies. For instance, UML case tools with support of rules may be used for modeling of Semantic Web applications, which include ontologies and rules.

Potential - the proposed method for visual rules modeling and verbalization introduces the possibility for rule–based software development, which is a powerful paradigm for special classes of software applications (for example: insurance, mortgage, business automation).

In this paper we have described a UML-based rule modeling approach. We have provided a quick overview of existing rule modeling technologies for business automation, software development and the Semantic Web and argue that our approach can be used for formalization of business requirements, which is demonstrated by means of two examples. The full specifications of R2ML and URML are available in the D8 deliverable of the REWERSE Working Group I1.

The modeling approach has been evaluated on several business case studies. The future work in this area is towards the approach evaluation on real business applications. The issue of modeling of reaction rules-driven Semantic Web Services is currently under consideration.

Acknowledgment

This research has been funded by the European Commission and by the Swiss State Secretariat for Education and Research within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

References

1. Object Constraint Language (OCL), Version 2.0, `http://www.omg.org/docs/ptc/03-10-14.pdf`
2. EU-Rent case study on I1 website `http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=node/12`
3. G. Klyne and J.J. Caroll (Eds.), Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C, 2004
4. Patel-Schneider, Peter F., Horroks I., OWL Web Ontology Language Semantic and Abstract Syntax, `http://www.w3.org/2004/OWL`
5. Patel-Schneider, Peter F., Horroks I., Boley H., Tabet S., Grosof B, Dean M., SWRL: A Semantic Web Rule Language Combining OWL and RuleML, `http://www.w3.org/Submission/2004/SUBM-SWRL-20040521`
6. Ross, R. G., Principles of the Business Rule Approach. Addison-Wesley Information Technology Series (2003).
7. Wagner G., Antoniou G., Tabet S., Boley H.: The Abstract Syntax of RuleML - Towards a General Web Rule Language Framework
8. Wagner, G., Giurca, A., Lukichev, S. (2006). A Usable Interchange Format for Rich Syntax Rules. Integrating OCL, RuleML and SWRL. Will appear in proceedings of Reasoning on the Web Workshop at WWW2006, May 2006
9. Meta Object Facility (MOF) Core Specification, Version 2.0, January 2006, `http://www.omg.org/docs/formal/06-01-01.pdf`
10. Unified Modeling Language (UML), version 2.0, `http://www.omg.org/technology/documents/formal/uml.htm`