

# K-Site Rules at the RuleML Challenge 2009

José L. Martínez-Fernández<sup>1,2</sup>, José C. González-Cristóbal<sup>2,3</sup>, Paloma Martínez<sup>2</sup>

<sup>1</sup> DAEDALUS – Data, Decisions and Language S.A.

Avda. de la Albufera, 321

28031 Madrid, Spain

{jmartinez, jgonzalez}@daedalus.es

<sup>2</sup> Advanced Databases Group, Universidad Carlos III de Madrid

Avda. de la Universidad, 30

28911 Leganés, Spain

{jose Luis.martinez, paloma.martinez}@uc3m.es

<sup>3</sup> DIT, Universidad Politécnica de Madrid

Avda. Complutense, 30

28040 Madrid, Spain

josecarlos.gonzalez@upm.es

**Abstract.** There exist a great number of business rules engines in the market, each one with its own rules language. As a result, the need of a standard rules definition language is obvious. In this demo paper, the application of a combination of standard business rules languages to develop a business rules management tool, K-Site Rules, is explained. The main goals of this tool are: first, to integrate the development process for business rules in the software development lifecycle and, second, to make this development independent of rule engine products. K-Site Rules does not include its own rule engine implementation but acts as a broker to the preferred rule engine. In this way, business rules can be reused across different rule engines (which can change because of economical issues, performance issues, etc.) and through different projects in the company. The K-Site Rules tool is part of ITECBAN, a project devoted to the definition of a new platform core for banking applications.

**Keywords:** Business Rules Management System, RIF, OWL, Model Driven Architecture

## 1 Introduction

In the last few years, companies have started to introduce business rules systems (BRS) as one of the components of their information systems. The ability to adapt business processes to changes in the environment with low development costs is the main reason to introduce this knowledge based systems in the company's IT infrastructures. If business rules are properly identified, business behavior can be modified without applications recoding.

According to the Business Rules Group<sup>1</sup> (BRG), a business rule can be defined as “*a directive, intended to influence or guide business behavior, in support of business policy that has been formulated in response to an opportunity, threat, strength, or weakness*”. From the Information Technology (IT) point of view a business rule can be defined as “*a statement that defines or constrains an aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business*”. This work is centered on the latter definition.

There are several main drawbacks in the integration of BRSs in the IT infrastructure of the company: first of all, it is not clear how to engage the software development cycle with the business rules approach since business rules are usually scattered in the application code. Second, there are a lot of commercial products implementing business rules but there is no common way to express business rules; this lack of widespread standards is a main issue in the business rules field.

Currently, there is a trend in software development based on the Model Driven Architecture (MDA) [3] where the main goals are promoting interoperability, portability and usability. For this purpose, several steps are taken: obtaining a system specification independent of the platform supporting it, identifying possible platforms, choosing a platform and transforming the generic model into the platform specific model. The right place in this architecture for BRSs must be envisaged to assure a good integration with the rest of technologies used by current IT systems. In this way, current drawbacks could be overcome. So, MDA assumes independence from the business rule engine to be used to implement the system. For this purpose, it can be assured that the principles of the Semantic Web can be applied, from the Ontology Web Language (OWL) [6] standard to the Rules Interchange Format (RIF) [7]. So, the ability to formalize the domain upon which business rules are defined through the use of ontologies can constitute an improvement for business rule systems.

This demo paper describes a prototype of K-Site Rules, the tool for business rule development that is being developed in the ITECBAN project (Architecture for Core Banking Information Systems), funded by the Spanish Center for the Development of Industrial Technology (CDTI, Ministry of Industry, Tourism and Trade), INGENIO 2010 Program.

## 2 K-Site Rules Framework Description

K-Site Rules is the name given to the tool developed by DAEDALUS and INDRA as part of the ITECBAN project. K-Site Rules is a tool specifically designed to support the business rules development process when rule engines are considered as part of the architecture of an IT system. Two main objectives have guided the development of K-Site Rules: first, independence of the rule engine selected to implement business rules, that is, the definition of the rule must be independent of the rule engine where it is going to be executed. This situation is depicted in Figure 1.

---

<sup>1</sup> <http://www.businessrulesgroup.org>

Secondly, an easy and simple integration in the software development cycle adopted in the target organization must be provided. For this reason, K-Site Rules developer editor is integrated in the ECLIPSE development environment<sup>2</sup>.

Figure 2 shows the software components of K-Site Rules, where three basic elements can be distinguished.

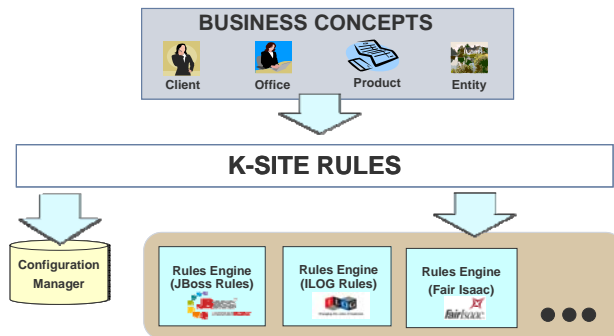


Figure 1. K-Site Rules as an intermediary framework

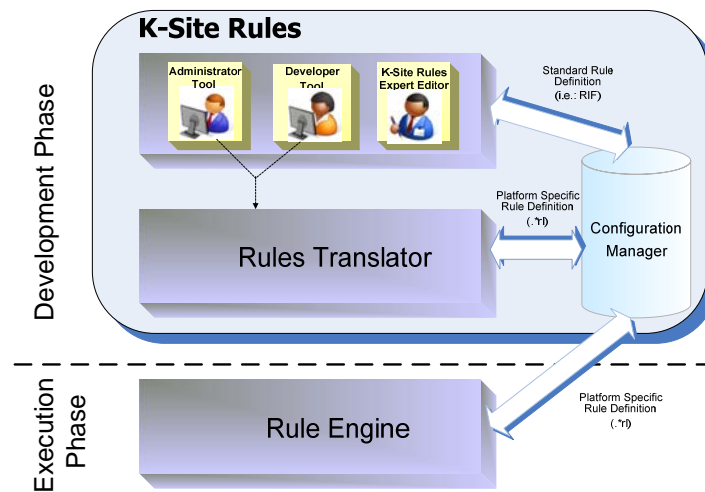


Figure 2. K-Site Rules software components architecture

### Rule Editors

K-Site Rules has been developed taking into account three different user roles: business analysts (also called domain experts), developers and tool administrators.

<sup>2</sup> <http://www.eclipse.org>

4 José L. Martínez-Fernández, Paloma Martínez, José C. González-Cristóbal

Business analysts are people with knowledge about the business but no technical capabilities, only as office software users. These people need tools with very simple user interfaces for them to be able to put their knowledge about the business in the form of business rules that can be processed by information systems. This is the main goal of the K-Site Rules Expert editor, allowing the definition of business rules using natural language, decision tables or decision trees in a web based interface. This module produces a standard definition of the business rules, currently using the RIF format which is stored in the configuration manager module. This web module also includes features to validate the defined business rules by providing test values and expected results when the rule is applied.

Developers, on the other hand, have a deep technical knowledge, but they are less aware about business domain concepts. For them there is also a particular tool, allowing the design, implementation and testing of business rules. This tool is integrated in the development environment, an Eclipse based development environment in the first implementation of K-Site Rules. As the K-Site Rules Expert editor does, the Developer Editor generates a standard expression of the business rule using RIF. This standard version is stored in the configuration Manager Module and is shared with business analysts. The developer has the ability to select the rule engine that will be used to execute the business rule, and to produce an automatic translation to the specific language of the selected rule engine. This module also provides the ability to validate business rules by giving test data and desired results.

Table 1 shows the technology environment selected for the implementation of both rule editors.

**Table 1.** Development technologies used in K-Site editors

Technical Feature	K-Site Rules Expert Editor	Development Editor
Base tool	Browser based	Eclipse based
Development technology	Tomcat 6 + Google Web Toolkit	JAVA Programming Language
Installation requirements	None	Eclipse + Plugin configuration

There is a component in charge of transforming Unified Modeling Language (UML) models to OWL ones, which are then graphically depicted in order to facilitate users understanding of the knowledge model. The steps for the transformation among models are taken from the Ontology Definition Metamodel (ODM) [4] specification. This transformation is implemented using JAVA programming language and a library to deal with OWL structures, OWLapi [2]

### **Rules Translator**

This is the core component of K-Site Rules and it is in charge of obtaining rule expressions according to a specific rule engine. The input to this module is a standard expression of the business rule, currently based on the RIF language. The output is a representation of the same business rule in the specific language understood by the rule engine. For example, if the rule engine selected to implement rules is ILOG JRules, the RIF representation must be translated to IRL, the rule language supported by ILOG JRules. If K-Site Rules is going to be included in a development process based on a Model Driven Architecture, this component is crucial to allow an automatic generation of an implementation of a system from the models defining that system.

There are different translation steps in the process. The first one is from RIF format to OMG Production Rules Representation (OMG PRR) [5] and the second one from OMG PRR to the proprietary language used in the rule engine that is supporting the Business Rules Management System. The RIF to OMG PRR translation is done following the operators defined at the RIF Production Rules Dialect (RIF PRD) standard. A specific software component has been implemented in K-Site Rules to perform this transformation.

### **Configuration Manager**

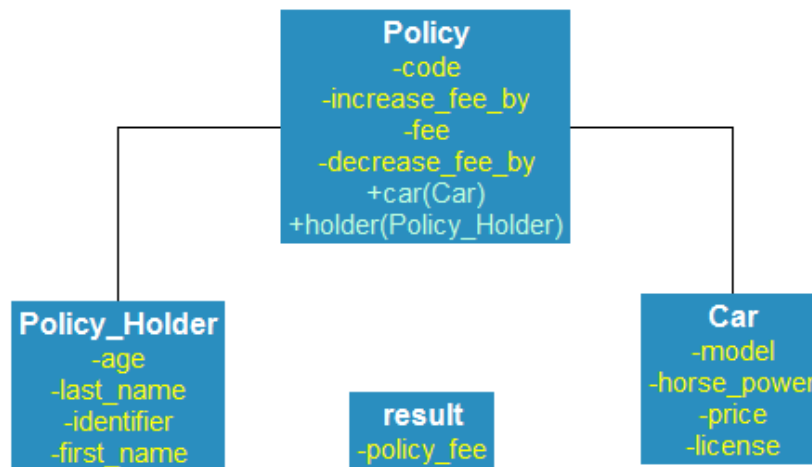
The Configuration Manager Module is a repository where business rules, probably with different versions, are stored. The current implementation of this module is based on Subversion, SVN [1], which provides needed version control and team work capabilities. In fact, developer and business analyst tools communicate through this component; changes made by one role to a business rule are shared with the other role through the configuration manager module. This component also allows auditing of business rules under development.

## **3 Software development cycle integration**

K-Site Rules considers business rules at every stage of the development process. At the analysis phase, K-Site Rules provides tools to allow business experts the definition of business rules using natural language expressions and their vocabulary. Business analysts even have the tools to test the functionality of the business rules they are working in. At design time, developers can review, test and validate business rules in the specific rule language (IRL, DRL, etc.) built by the tool, by accessing the business rules expressions stored in the configuration management system. If it is needed they can develop additional business rules or adapt obtained implementations. Regarding business rules deployment, K-Site Rules does not impose any restriction, business rules produced by the tool reside at the configuration management system and can be deployed according to deployment policies defined at the company.

## 4 Example

This section summarizes the business rules development process using K-Site Rules expert editor. Suppose a car insurance company where conditions to give a policy to a customer can be expressed using rules. This is the domain model defined for the example environment shown at the test web site (please, follow section 5 instructions to access), at the Business Model tab. Figure 3 depicts the diagram for the insurance company model.



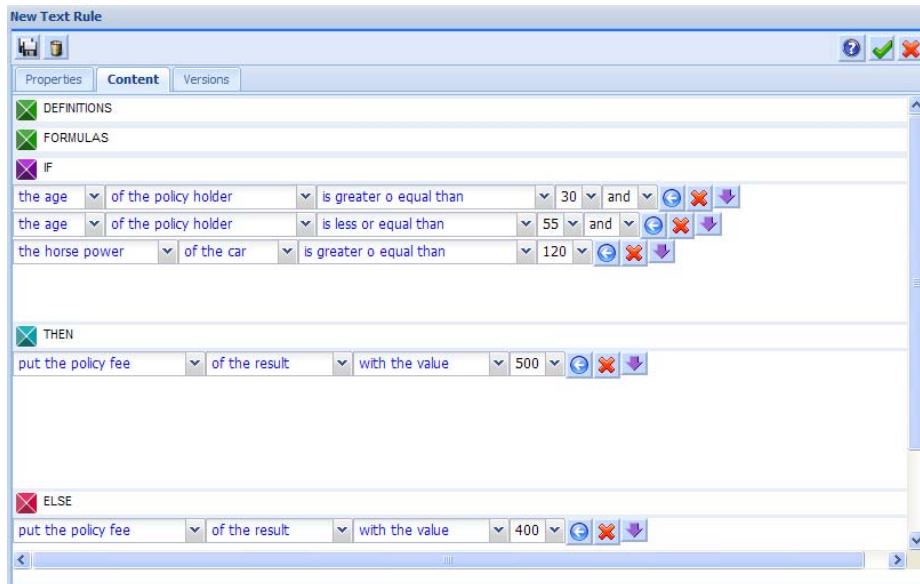
**Figure 3.** Business model shown by K-Site Rules expert editor

Three main concepts are used in the example, the policy holder (Policy\_Holder), who is the customer contracting the policy, the car involved in the policy (Car) and the policy itself (Policy).

Suppose that you want to define the following business rule to state the fee that a customer must pay for an insurance:

*If the customer is between 30 and 55 years old and the rated horse power of the car to include in the policy is over 120 HP then the fee to pay is 500\$. In any other case, the fee will be 400\$*

This business rule will be part of a decision service devoted to the definition of the fee that each customer must pay. The input for the decision service is formed by three concepts: the policy to be contracted (a Policy), data for the holder of that policy (a Policy Holder), and data of the car involved in the policy (Car). It is possible to define the given business rule as a written rule using the corresponding tool provided by K-Site Rules expert editor, once a ruleset has been defined. The written rule obtained is depicted in Figure 4.



**Figure 4.** Written business rules definition with K-Site Rules expert editor

Once the rule is defined, a validation process can be done through the functionality included in the Validation tab. At this tab, some example data can be given for instances of the concepts appearing as the input of the decision service.

Of course, validation data can be loaded through Excel files and, for developers, there is also the possibility of using serialized objects.

## 5 Conclusions

This demo paper is supported by the presentation available at <http://clients.daedalus.es/MK-2009-43-ITECBAN-01.pdf>. This presentation includes a guided tour through the different editors, describing the steps to be given in order to develop a business rule. There is also the possibility of testing K-Site Rules Expert tool, using as credentials: 'ruleml' as the user and 'ruleml09' as the password, through the following URL: <http://163.117.129.57:8080/KSiteRulesExperto/?locale=en> On the other hand, the development tool must be installed on a local machine so no web access can be provided but local demos can be performed if required.

## ACKNOWLEDGEMENTS

This paper has been partially supported by the Spanish Center for the Development of Industrial Technology (CDTI, Ministry of Industry, Tourism and Trade), through the project ITECBAN (Architecture for Core Banking Information Systems), INGENIO 2010 Programme. Other partners in ITECBAN are INDRA Sistemas, CajaMadrid,

Sun Microsystems and Grid Systems. Special mention to our colleagues at INDRA must be done for their involvement in the specification of K-Site Rules: Fernando Alcántara, Pablo Leal, Juan Carlos Macho and Gonzalo Pando (in alphabetical order).

## REFERENCES

1. Collins-Sussman, B., Fitzpatrick, B.W., Pilato, C.M. Version Control with Subversion: For Subversion 1.6, Available at <http://svnbook.red-bean.com/nightly/en/svn-book.pdf>, last visit: 11/03/2008, (2007)
2. Horridge, M., Bechhofer, S., Noppens, O. Igniting the OWL 1.1 Touch Paper: The OWL API. OWLED 2007, 3rd OWL Experienced and Directions Workshop, Innsbruck, Austria, (2007)
3. Miller, J., and Mukerji, J., *MDA Guide Version 1.0.1*, OMG. Available at <http://www.omg.org/docs/omg/03-06-01.pdf> (2003)
4. OMG, Ontology Definition Metamodel (ODM). Available as ptc/2008-09-07 at <http://www.omg.org/docs/ptc/08-09-07.pdf>, last visit: 17/02/2009, (2005)
5. OMG, Production Rules Representation (PRR), Beta 1. Available as ptc/2008-09-07 at <http://www.omg.org/spec/PRR/1.0>, last visit: 17/02/2009, (2006)
6. *OWL Web Ontology Language Semantics and Abstract Syntax*. Peter F. Patel-Schneider, Pat Hayes and Ian Horrocks, eds. W3C Recommendation 10 February 2004. Latest version is available at <http://www.w3.org/TR/owl-semantics/> (2004)
7. Sainte Marie, Christian de, *RIF Production Rules Dialect (RIF-PRD)*. Available at <http://www.w3.org/TR/2008/WD-rif-prd-20080730/>, last visit: 10/06/2009, (2008)