

# Business Rules Languages and Model-Interoperability

## (Position Paper)

Devasis Bassu and Ashish Jain

*Applied Research*

*Telcordia Technologies*

*One Telcordia Drive*

*Piscataway, NJ 08854*

{dbassu, jain}@research.telcordia.com

### **Abstract**

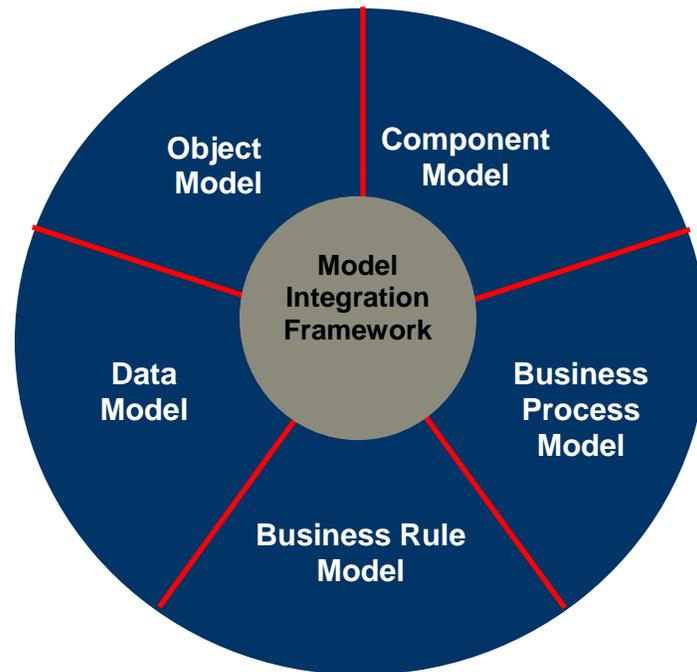
*Declarative Business Rules provides a framework for making software components flexible. Since several key business process activities are implemented using software components, one can argue that business rules technology can help in achieving business agility. Business rules raise the level of abstraction of specifications, making it possible for a business analyst to specify the rules which can be automatically translated to executable code. However, based upon our considerable experience in using business rules technologies for developing mission-critical Telecom Operation Support systems, we believe that major interoperability issues need to be addressed before the 'on-demand' flexibility vision in Information Systems can be realized. The position paper highlights some of those issues and suggests some approaches.*

## **1. Introduction**

Business rules are present in almost every aspect of telecommunication systems. There are rules for ordering, planning, provisioning, activating, monitoring, work-force scheduling, and so forth. Business rules also appear as specifications for messages exchanged between two systems. Enterprise Architectures, such as Zachman framework [1,2], are a collection of models. Additionally all of the model artifacts should provide appropriate views for the various stakeholder needs. For example, a business rule implemented as SQL trigger is not an appropriate abstraction for a business analyst. It is also extremely important that business rules be shared and understood across companies and even among different business groups within a company. The large volume of these rules and frequent changes to them add to the complexity of disseminating the information at the *right level of abstraction* to diverse groups within a reasonable amount of time.

Traditionally, unstructured prose has taken the place of providing the views. However, business rules models offer considerable promise in streamlining the process of authoring, collaborating, maintaining, and testing. One of the most important challenges in building enterprise level systems today is the *management of information* (requirements, data, rules, etc.) across tiers of an enterprise. One of the major reasons which make management of this information hard is the need to change rules and models very frequently. There is a growing demand for building open and flexible systems which will not only use standardized languages/protocols but also allow 'in-the-field'

customization and extension of an application by a client. Traditional Application Lifecycle management tools and solutions do not provide the needed integration between the models depicted in Figure 1. This prevents from consistent editing of information by an end-user. Examples of consistent upfront editing are a) preventing a user from defining a business rule on non-existent objects, or b) preventing users from specifying rules which are violated in the object model.



**Figure 1: Model Integration Framework**

## ***2. Our approach***

Central to the specification of business rules in any enterprise system is (are) the business data model(s). Recognizing the need for providing different views of the same information (data models, rules, etc.), all information is stored in a standard format. Role based views and editors are provided to distribute and edit the information. This approach is in accordance with the principles of the Zachman framework for enterprise architecture. Since the core information is structured, we also provide translators/compiler which can then generate code. The models are created using the Eclipse Modeling Framework (EMF) [4]. We use the Eclipse meta-IDE framework to provide an integrated modeling and development environment which allows different perspective based upon the owner. One of our major interests in participation is to discuss the interoperability issues between Business Rules Models based upon MOF and other MOF based languages like UML.

As an example, the data model for the specification of business rules may be the W3C XML schemas derived from the UML business object model. For Telcordia's business analysts, we have defined an EMF based business rules model which interoperates with

the UML 2.0 models. We use appropriate Eclipse plugin editors to edit the information at the needed level of abstraction, making sure that cross model dependencies are always maintained. For example, one of the plugin editors provides plain English based editing view with incremental editing checks. A declarative and extensible meta-model is used to represent telecom specific vocabularies (verbs and actions). Using an XMI serialization of the rules model, we obtain the XML representation which can be compiled into runtime XSLT scripts, ILOG JRules, or Java code.

### **3. Other related issues**

As mentioned before, business rule base can become large. Categorization rules packages based upon use case model is the most natural organization for the rules. This makes rule packaging and its integration with UML use case models very important. Traceability to requirements and/or use-cases is necessary to account for fulfillment of all user requirements.

Another issue is the error codes and error strings. It's our experience that majority of the B2B (Business to Business) interfaces contain rules with associated error object. If a business process fails, the partner need a plain-text error message with an error code explaining the failure. The meta-model for business rules language need to have an explicit support for error objects.

We believe that business rules framework have a tremendous potential in reducing the total cost of ownership of a software solution by providing flexibility and ease of use for business analysts. We have been able to successfully use business rules technologies in large-scale systems where the business analysts were able to directly and safely configure and extend the desired behavior of a fielded-system. However, it is our position that addressing the issues listed below is important for practical adoption of business rules systems, and we believe that these issues should be in the scope of any Business Rules standardization efforts.

- Allow for flexibility in referencing the underlying data model(s). For example, the data model may be specified using XML schema(s) or UML (XMI).
- Allow for different views of the rules, i.e., domain-specific and/or user-defined syntax mappings for the rule language.
- Ability to restrict/extend the rule language, i.e. allow/disallow certain rule constructs for specifying rules for a particular domain.
- Support notion of a rule package which allows specification of global rules.
- Ability to import rules from other rule packages – rule package reuse.
- Allow for associated metadata with a rule and a rule package.
- Allow special processing instructions – to be used by processors which work on the rules.

## References

1. "A Framework for Information Systems Architecture." John A. Zachman. IBM Systems Journal, vol. 26, no. 3, 1987. IBM Publication G321-5298.
2. "Extending and Formalizing the Framework for Information Systems Architecture." J.F. Sowa and J. A. Zachman. IBM Systems Journal, vol. 31, no. 3, 1992. IBM Publication G321-5488.
3. The Zachman Institute for Framework Advancement. <http://www.zifa.com>.
4. The Eclipse Modeling Framework, Eclipse Foundation. <http://www.eclipse.org/emf>.
5. Meta-Object Facility, Object Management Group. <http://www.omg.org/mof>.
6. Unified Modeling Language, Object Management Group. <http://www.uml.org>.
7. UML 2.0 Object Constraint Language, Object Management Group. <http://www.omg.org/cgi-bin/doc?ptc/2003-10-14>.
8. RuleML, the Rule Markup Initiative. <http://www.ruleml.org>.
9. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. <http://www.daml.org/2003/11/swrl>.
10. Jess, the Rule Engine for the Java Platform. <http://herzberg.ca.sandia.gov/jess>.
11. ILOG Business Rules – JRules. <http://www.ilog.com/products/jrules>.

**Background:** *Telcordia Technologies is a leading provider of Software and Services for IP, PSTN, Cable, and Wireless communication networks. Telcordia's annual revenues are close to a billion dollars and it employs approximately 3200 employees. Telcordia is the North American market leader for Telecom Operations and Business Support Systems. Telcordia is headquartered in Piscataway, NJ.*