Towards a Model Driven Semantics using the Ontology

Kaninda Musumbu

LaBRI (UMR 5800 du CNRS)
351, cours de la Libération, France - 33.405 Talence Cedex

Abstract

The Semantics Web is a vision for the future of the Web in which information is given explicit meaning, making it easy for machine to automatically process and integrate information available on the Web. Ontologies are used by people, database and applications for sharing domain information (a domain is a specific area of knowledge like medicine , financial management, etc.). The main goal of this paper is to present comprehensive introduction into MDA based ontology development. It will provides in introduction to the field of the Semantics Web and ontology engineering.

Keywords: Business rules, knowledge based systems, Model Driven Architecture, reasoning.

1. Introduction

The Semantics Web is the main direction of the future Web development. Domain ontologies are the most important part of Semantics Web applications. Artificial intelligence techniques are used for ontology creation, but those techniques are more related to research laboratories. Recently, there are many proposals to use software engineering techniques, especially the UML since it is the most accepted software engineering standard, in order to bring ontology development process closer to wider practitioners' population. However, UML is based on object oriented paradigm, and has some limitation regarding ontology development. These limitations can be overcome using UML's extensions (i.e. UML profiles), as well as other OMG's standards (i.e. Model Driven Architecture - MDA). Currently, there is an initiative (i.e. RFP) within the OMG aiming to define a suitable language for modeling Semantics Web ontology languages in the context of the MDA.
The Semantics Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for the Semantics Web is an ontology language what can formally describe the meaning of terminology used in Web documents. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema. The OWL Use Cases and Requirements Document provides more details on ontologies, motivates the need for a Web Ontology Language in terms of six use cases, and formulates design goals, requirements and objectives for OWL.

The aim of this paper is to give the methodology to automatically generate a part of the business rules by combining Model Driven Architecture and the Semantics Web using Ontology Definition Meta-model. This paper will be divided in three parts. The first covers the basis of both main topics ontology, semantics web and standards.

The second part, explain the Model Driven Architecture. The last is the central one, it starts with a review of several approaches and aim to bridge the gap between ontology development and software engineering methodologies.

2. The Ontology concept

The term ontology means a specification of a conceptualization. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. In that context, an ontology is a specification used for making ontological commitments. The formal definition: "An ontology defines the terms used to describe and represent an area of knowledge”.

Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them (note that here and throughout this document, definition is not used in the technical sense understood by logicians). They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable.

The Semantics Web needs ontologies with a significant degree of structure. These need to specify descriptions for the following kinds of concepts:

- Classes (general things) in the many domain of interest.l
- Relationship that can exists among things
- Properties (or attributes) those things may have

Ontologies are usually expressed in a logic-based language, so that detailed, accurate, consistent, sound, and meaningful distinctions can be made among the classes, properties, and relations. Some ontology tools can perform automated reasoning using the ontologies, and thus provide advanced services to intelligent applications such as conceptual/semantics search and retrieval, software agents, decision support, speech and natural language understanding, knowledge management, intelligent databases, and electronic commerce.

Ontologies figure prominently in the emerging Semantics Web as a way of representing the semantics of documents and enabling the semantics to be used by web applications and intelligent agents. Ontologies can prove very useful for a community as a way of structuring and defining the meaning of the metadata terms that are currently being collected and standardized. Using ontologies, tomorrow's applications can be "intelligent," in the sense that they can more accurately work at the human conceptual level.

Ontologies are critical for applications that want to search across or merge information from diverse communities. Although XML DTDs and XML Schemas are sufficient for exchanging data between parties who have agreed to definitions beforehand, their lack of semantics prevent machines from reliably performing this task given new XML vocabularies. The same term may be used with (sometimes subtle) different meaning in different contexts, and different terms may be used for items that have the same meaning. RDF and RDF Schema begin to approach this problem by allowing simple semantics to be associated with identifiers. RDF Schema is a simple ontology language. However, in order to achieve inter-operation between numerous, autonomously developed and managed schemas, richer semantics are needed. For example, RDF Schema cannot specify that the Person and Car classes are disjoint, or that a string quartet has exactly four musicians as members.

3. The Model Driven Architecture

The Model-Driven Architecture (MDA) starts with the well-known idea of separating the specification of the operation of a system from the details of the way that system uses the capabilities of its platform (PIM). In theories, the latter models must be partially generated from the form er. Platform independent models must be permanent, i.e. they do not contain any information about execution platform (is it a J2EE or .NET etc. application).
For constructing the concrete application, we must have platform specific models (PSM). These models are obtained by transforming PIM and adding technical information relative to platforms. PSM are not permanent models. All these models are for facilitating code generation. The MDA approach is widely used and advanced generators exist.

3.1MDA models and semantics

MDA principals are very interesting and allow economizing many times during application life cycle by code and model generation. However, MDA specification does not tell anything about semantics on models. MDA is only interested by content and not context then that using semantics will offers more interesting way in automatic generation. Making transformations between CIM and PIM, between PIM and PSM, and between PSM and code are done by specifying transformations rules. Nowadays these rules are manuals and machines cannot generate it because there is no notion of semantics between the entities that are concerned by transformations.
Business rules are about meanings and act on models. Generating all business rules is impossible but it would be possible to generate a large set of generics among them.

3.2 Solutions for adding semantics in models

In MDA, an instance of MOF is use for representing models but our works are only concerned by UML models. For adding semantics in UML models we can use:

- UML Profile: can be used for modeling many domains. The problem with this is that UML models are so generic that it is impossible to know either it is object application, a meta-model, a model, a database structures or anything else only by looking at. But this is not exploitable by machines because there is no notion of logic and taxonomy and semantics is not formally defined.

- Object Constraint Language: In UML it was not possible to define the body of an operation (or a method) so the OCL was standardized by OMG for doing it. OCL allows expressing any kind of constraints on UML models. For example, we can express constraints.

- Action Semantics: remember that the main constraint with OCL was that he only supports no side effects operations. To solve this constraint, the OMG standardize Action Semantics. Well, now we have a formalism being able to express any kind of operations and constraint but it is not enough.

As we can see, none of the UML “techniques” is suitable for adding semantics in models. In another side a new domain of computer is growing more and more: semantics web. The aim of the semantics web is to make the web both comprehensible by humans and machines. A part of semantics web is about ontology and reasoning. Modeling concept defined by ontologies can be used to model the concepts in a domain, the relationships between them, and the properties that can be used to describe instances of those concepts . In addition, the Web Ontology Language (OWL) supports the inclusion of certain types of constraint in ontology, allowing new information to be deduced when combining instance data with these logic's description . At this point our dilemma was how can we use MDA models and Semantics Web? Ontology Definition Meta Model was the response to our need.

3.3The Ontology Definition Meta-model

The MDA and its four-layer architecture provide a solid basis for defining the Meridel's of any modeling language, and thus a language for modeling ontology based on the MOF. The ODM is a proposal for an OMG's RFP (Request For Proposal) resulting of an extensive previous research in the fields of the MDA and ontology. The main objective of the ODM is to bridge the gap between traditional software tools for modeling (like UML) and artificial intelligence techniques (Logic's description) for making ontology. The principle of ODM is to merge two big domains of research which are Model Driven Architecture and Semantics Web.

4. Adding semantics on models for automatic business rules generation

MDA technologies and Semantics web are complementary; the first is concerned about automating the physical management and interchange of mandate, while

5. Conclusion

A business rules application is intentionally built to accommodate continuous change in business rules. The ability to change them effectively, are fundamental for improving business adaptability. The platform on which the application runs should support such continuous change. Offering to knowledgeable business people (experts) the possibilities to formulate, validate, and manage rules in a “zero-development” environment bring more value-added to this notion of “computer sciences in humanity's service”.
Allowing an automatic generation part of this business rules will be better.
In this paper, we have seen that, by combining the two big domains, Model Driven Architecture and Semantics Web, a solution is possible. Adding semantics on conceptual models will open an exciting and interesting domain of application like information merge.

6. References

[1] Barbara von Halle : “*Business Rules Applied”* John Willey & Sons, New York, USA, 2002

[2] Ronald G Ross : “*Principle of the Business Rule Approach”* Addison-Wesley, Boston, USA, 2003

[3] Xavier Blanc: “*MDA en action”*
Eyrolles, France, 2005

[4] The Object Management Group OMG : “*Unified Modelling Langauge Superstructure.”* OMG , February 2004

[5] Dragen Gaevie, Dragen Djurie and Vladan Devedzie: “*Model Driven Architecture and Ontology Engineering”* Springer-Veralg, Berlin,DE, 2006.