HAL
open science

Towards a Model Driven Semantics Web using the Ontology

Kaninda Musumbu

To cite this version:

Kaninda Musumbu. Towards a Model Driven Semantics Web using the Ontology. The 2013 International Conference on Advanced ICT for Business and Management (ICAICTBM2013), Sep 2013, Sanya, Hainan, China. pp.700. hal-00959977

HAL Id: hal-00959977

https://hal.science/hal-00959977v1

Submitted on 17 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Model Driven Semantics Web using the Ontology

K. Musumbu
LaBRI (UMR 5800 du CNRS),
351, cours de la Libération, F-33.405 TALENCE Cedex
e-mail: musumbu@labri.fr

Abstract

The Semantics Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). The main goal of this paper is to present comprehensive introduction into MDA-based ontology development. It will provide an introduction to the field of the Semantics Web and ontology engineering, a description of several UML- and metamodeling- based solutions and tools for ontology development, an overview of the OMG's MDA effort and related standards (Meta-Object Facility -MOF, UML, XML Metadata Interchange -XMI).

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

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.

Business rules define and constrains business processes in enterprises. Therefore, many business-governing rules have to be implemented in business-supporting applications, in order to reflect the real business environment. 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. We will also discuss the possibilities and the benefits provided by mixing models and web reasoning.

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 comm unity of agents.dixit T. R. Gruber [17].He .has been designing ontol ogies for the purpose of enabling knowledge sharing and reuse. In that context, an ontology is a specification used for making ontological commitments. The form al definition of ontological commitment is given below."

An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.).

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 word ontology has been used to describe artifacts with different degrees of structure. These range from simple taxonomies (such as the Yahoo hierarchy), to metadata schemes (such as the Dublin Core), to logical theories. 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 domains of interest
- The relationships that can exist among things
- The 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. With RDF Schema, one can define classes that may have multiple subclasses and super classes, and can define properties, which may have sub properties, domains, and ranges. In this sense, 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.

One of the goals of this document is to specify what is needed in a web ontology language. These requirements will be motivated by potential use cases and general design objectives that take into account the difficulties in applying the standard notion of ontologies to the unique environment of the Web.

3 The Model Driven Architecture (MDA)

The Model-Driven Architecture starts with the well-known and long established 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 [8]. The Figure 1 gives a general view of the MDA approach. We can see that a constru ction of a new Information System begins with the development of one or more req uirements models (CIM). Then we may develop models independent from any platform (PIM). In theories, the latter models must be partially generated from the form er. Platform independent models must

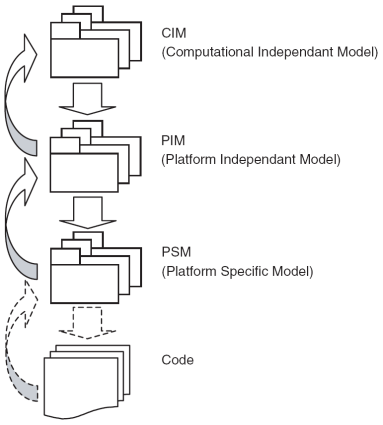

Figure 1: Global view of the Model Driven Architecture approach

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 infor mation 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.

3.1 MDA models and semantics

MDA principals are very interesting and allow economizing many times during appl ication 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 au tomatic generation.

3.1.1 Why should MDA talk about semantics:

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 rul es is impossible but it would be possible to generate a large set of generics am ong them. For example for the model in Figure 2, we should want to generate business rules like:

- Every *Human* must have a father and a mother.

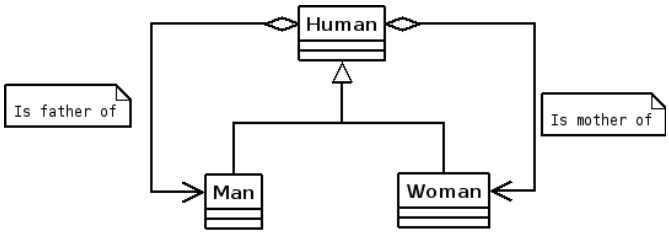

Figure 2: A little human model

- IF a *Human* is the mother of a *Human* then this *Human* is a *Woman*.
- IF a *Human* is the father of a *Human* then this *Human* is a *Man*.

For doing this automatically, it is clear that adding semantics in models is needed.

3.1.2 Eventual solutions for adding semantics in models:

In MDA, an instance of MOF [10] 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*: UML 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 it [9]. For adding precision, the OMG has standardized the concept of UML profile [11]. A UML profile is a set of techniques and mechanisms allowing to adapt UML in a particular and specific domain. UML profile can be used in any UML model and do not modify the structure of the meta-model. UML profiles are stereotypes or labels which can be injected into models. After having stuck labels on models, we can make inference using it. As we can see, doing this can solve our problem of semantics lack on model in a low level, 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 [12] was standardized by OMG for doing it. OCL allows expressing any kind of constraints on UML models. For example, we can express constraints like: "before renting a car you must be sure that it is OK". Well, OCL seems

to be a good solution for our problem but it is not the case. The first problem with OCL is that he does not support side effect operations and the second is that he does not offer automatic inference for machines.

- *Action Semantic*: 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 [13]. Well, now we have a formalism being able to express any king of operations and constraints but it is not enough. This formalism is complicated to use [9], was not created while thinking to machine comprehension and self-use, and do not have a textual formalism.

As we can see, none of the UML "techniques" is suitable for adding semantics in models. In an other 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 [14]. 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 [15]. In addition, the Web Ontology Language (OWL)[24] 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 [15]. 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.

3.2 The Ontology Definition Metamodel

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 [16]. The ODM is a proposal for an OMG's RFP (Request For Proposal) [18] resulting of an extensive previous research in the fields of the MDA and ontology [19, 20]. 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. ODM is still in standardization process at the OMG [21] when this paper is being written. Basically the ODM allows making ontology using UML (by using an UML refile with existing tools like Rational Rose or Poseidon) and transforming it to OWL/RDF, Topic Map or Common Logic (Figure 3).

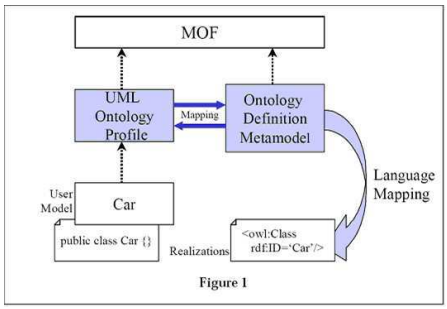

Figure 3: ODM principle

In next section, we will see how ontology reasoning can be used to solve the lack of semantics in models.

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 the latter is focused on the semantics embodied in the content of the mandate as well as on automated reasoning over that content [22]. The Semantics Web is the new-generation Web that tries to represent information such that it ca be used by machines not just for display purposes, but for automation, integration, and reuse across applications [23]. Model Driven Development (MDD) is being developed in parallel with the Semantics Web [24]. Emerging applications in finance, health-care, security, communications, business intelligence, and many other vertical markets are content and context sensitive (semantics), and require Enterprise scalability and performance [22]. Merging Semantics Web and MDA technologies can fill this lack. Merging these two domains will be benefice to both:

- MDA is only interested by content and not by context (semantics), semantics web will resolve this important problem.

- For semantics web: there are many existing ontologies development tools, and they are used by different groups of people for performing diverse tasks. Although each tool provides different functionalities, users tend to use just one tool, as they are

not able to exchange their ontologies from one tool to another. Supporting a heterogeneity of modeling language, while providing standard representations and APIs for model repositories and other tools, is one of the aims of the MDA (XMI) [15]. Another interesting thing is that so mature UML tools cold be used for making ontologies rather than using so theoretical languages from Artificial Intelligence domain.

Merging MDA and Semantics Web technologies allow more automatic processing like: generation of constraints and business rules from models.

4.1 Our Approach for business rules automatic generation

Our principal is to use the benefice and advanced researches in Semantics Web, to

combine it to Model Driven Architecture in the goal of making automatic business rules generation.
For generating business rules automatically, will use principally the semantic in OWL format. In OWL reasoning, we can make automatic reasoning both with structures (TBox) or assertion on individuals and properties (ABox) [25]. In our case for example, if we have:

$$Predicate : Human \longrightarrow Woman$$

This declaration means that we have a properties *Predicate* going to domai n *Human* to range *Woman*.
So we want to generate that:

$$\begin{array}{l} IF \\ \quad Object1\ Predicate\ Object2 \\ THEN \\ \quad Object1\ is\ of\ type\ Human \\ \quad ET \\ \quad Object2\ is\ of\ type\ Woman \end{array}$$

The Figure 4 schematizes our approach: using ODM, our model is generated in OWL/RDF model and this last one is enriched with semantics. With this semantically rich model two solutions are possible for generating something with: serialized the rich model in XMI [26] and use something like JMI [27] for parsing manually the XMI. Another solution is making inference directly with the OWL model using a OWL reasoner. We adopter the last solution because it exist good OWL Reasoner and this solution uses less intermediary steps.
Note that we know that generating any kind of business rules is impossible but if we arrive to generate a part of them, it will be a good thing. The Figure 5

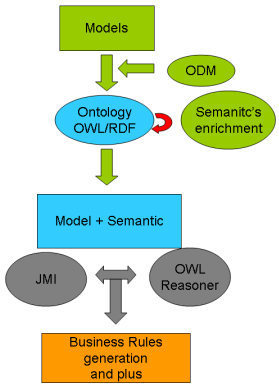

Figure 4: Our approach

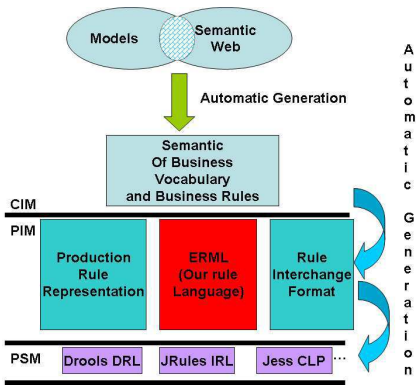

Figure 5: Our approach throughout the MDA layers

summarizes our approach throughout MDA layers. As we can see the first step will be a generation according to the Computation Independent Model (CIM) in a OMG SBVR like syntax (in natural language), and after this, the next step will be to generate executives rules according to the Platform Independent Model using our rule language [28] and models based on something like XMI. We can also see that in the figure, our business rules language or a future business rule formalism [4, **?**, 7], will be used to formalize rules generated at the PIM Level. At this step we'll use our "translators" for generating rules at the PSM level for a specific rule engine. If one day, a standard business rules language is adopted, we'll either make a "translator" from our language towards the new standard or either store directly rules in the new formalism.

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.

Right now we can only make generation according to the Computation Independent Model (CIM) in a OMG SBVR like syntax (in natural language). The next step will be to generate executives rules according to the Platform Independent Model using our rule language [28] and models based on something like XMI. The last step will be to have an editor allowing to edit both models and semantics.

Making simples generic business rules generation possible from models will facilitates the use of the business rules approach which allows easier systems maintenance. It's clear that generating all kind of business rules is an utopia and we must delimit the degree of generation we want to obtain.

Adding semantics on conceptual models will open an exciting and interesting domain of application like information merge.

References

[1] Barbara von Halle. *Business Rules Applied.* John Wiley & Sons, New York, USA, 2002.

[2] Ronald G. Ross. *Principles of the Business Rule Approach.* Addison-Wesley, Boston, USA, 2003.

[3] The Object Management Group OMG. *Semantics of Business Vocabulary and Business Rules (SBVR).* OMG Specification, March 2006.

[4] W3C. *Rule Interchange Format (RIF).* W3C Workgroup, 2005.

[5] The Object Management Group OMG. *Production Rule Representation (PRR) RFP.* OMG Request For Proposal (br/2003-09-03), 2003.

[6] IBM T.J. Watson Research Center. *CommonRules project.* Intelligent Agents project (1994-97), 1997.

[7] RuleML. *The RuleML initiative.*

[8] The Object Management Group OMG. *Model Driven Archtecture Guide Version 1.0.1.* OMG Specification, June 2003.

[9] Xavier Blanc. *MDA en action.* Eyrolles, France, 2005.

[10] The Object Management Group OMG. *Meta Objet Facility (MOF) Specification Version 1.4.* OMG Specification (formal/02-04-03), April 2002.

[11] The Object Management Group. *Unified Modeling Language: Superstructure.* OMG Specification, February 2004.

[12] The Object Management Group OMG. *UML 2.0 OCL Specification.* OMG Specification, October 2003.

[13] The Action Semantics Consortium. *Action semantics for the uml.* OMG Specification (ad/2001-03-01), March 2001.

[14] Thomas B. Passin. *Explorer's guide to the Semantics Web.* Manning Publications Co, Greenwich, UK, 2004.

[15] Stephen Cranefield and Jin Pan. *Bridging the Gap Between the Model-Driven Architecture and Ontology Engineering. Proc. of AOSE 2004 Workshop*, 2004.

[16] Dragan Gaěvič, Dragan Djurié, and Vladan Devedžić. *Model Driven Architecture and Ontology Development.* Springer-Verlag, Berlin, DE, 2006.

[17] Gruber, Thomas. Toward Principles for the Design of Ontologies Used for Knowledge Shar ing. International Journal Human-Computer Studies Vol. 43, Issues 5-6, Nove mer 1995..

[18] The Object Management Group OMG. *Request For Proposal for Ontology Definition Metamodel*. OMG Request For Proposal, March 2003.

[19] Kenneth Baclawski, Mieczyslaw K. Kokar, Paul A. Kogut, Lewis Hart, Jeffrey Smith, William S. Holmes III, Jerzy Letkowski, and Michael L. Aronson. *Extending UML to Support Ontology Engineering for the Semantics Web. Lecture Notes in Computer Science*, 2185:342+, 2001.

[20] Dragan Djuric, Dragan Gasevic, and Vladan Devedzic. *Ontology Modeling and MDA. Journal of Object Technology*, 4(1):109–128, 2005.

[21] The Object Management Group OMG, IBM, and Sandpiper Software. *Ontology Definition Metamodel*. OMG Specification, June 2006.

[22] H. Knublauch. *Ontology-Driven Software Development in the Context of the Semantics Web: An Example Scenario with Protege/OWL. 1st International Workshop on the Model-Driven Semantics Web (MDSW2004)*, 2004.

[23] Harold Boley, Said Tabet, and Gerd Wagner. *Design Rationale for RuleML: A Markup Language for Semantic Web Rules.* In *SWWS*, pages 381–401, 2001.

[24] Stephen J. Mellor, Anthony N. Clark, and Takao Futagami. *Guest Editors' Introduction: Model-Driven Development. IEEE Software*, 20(5):14–18, 2003.

[25] Raphael Volz. *Web Ontology Reasoning with Logic Databases*. PhD thesis, Universität Karlsruhe (TH), Universität Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe, 2004.

[26] The Object Management Group OMG. *MOF 2.0/XMI Mapping Specification, v2.1.* OMG Specification (formal/05-09-01), 2005.

[27] Java Community Process(JCP). *Java Metadata Interface (JMI).* Sun Java Specification Request (JSR 40), 2002.

[28] Mouhamed Diouf, Kaninda Musumbu, and Sofian Maabout. *Standard Business Rules Language: why and how? The 2006 International Conference on Artificial Intelligence*, June 2006.