



HAL
open science

Business Rules Generation methods by Merging Model Driven Architecture and Web Semantics

Kaninda Musumbu, Mouhamed Diouf, Sofian Maabout

► **To cite this version:**

Kaninda Musumbu, Mouhamed Diouf, Sofian Maabout. Business Rules Generation methods by Merging Model Driven Architecture and Web Semantics. 2010 First IEEE International Conference On Software Engineering and Service Sciences, Jul 2010, Beijing, China. pp.33-36. hal-00637872

HAL Id: hal-00637872

<https://hal.science/hal-00637872>

Submitted on 3 Nov 2011

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.

Business Rules Generation methods by Merging Model Driven Architecture and Web Semantics

Kaninda Musumbu
LaBRI and
University Bordeaux 1
351 cours de la Liberation
f-33405 Talence CEDEX
France
Email: musumbu@labri.fr

Mouhamed Diouf
LaBRI
351 cours de la Liberation
f-33405 Talence CEDEX
France
Email: moustaphadiouf@gmail.com

Sofian Maabout
LaBRI and
University Bordeaux 4
Avenue Leon Duguit
33608 Pessac CEDEX
France
Email: maabout@labri.fr

Abstract—The increasing complexity of the information systems must be taking into account for new technologies, and the appearance of new types of requirements raise new problems that the traditional engineering approaches of the information systems cannot always solve in an adapted way. Business Rules constitute a key element of the Semantic Web vision, allowing integration, derivation, and transformation of data from multiple sources in a distributed, transparent and scalable manner. The aim of this paper is to propose a way to automatically generate a part of the business rules by combining concepts coming from Model Driven Architecture and Semantic Web using the Ontology Definition Metamodel.

I. INTRODUCTION

Business rules are statements that express (certain parts of) a business policy, defining terms and defining or constraining the operations of an entreprise, in a declarative manner [1], [3]. The business rule approach is more and more used due to the fact that in such systems, business experts can maintain the complex behavior of their applications in a “zero development” environment. Currently the main need in this domain is having a standard language for representing business rules, facilitating their integration and share. Works for solving this lack is in progress at OMG and W3C as well as other initiatives.

In another side, an enough heavy step during business rules bases systems implementation is the step of elicitation of rules from the business. Entreprises, generally, have (legacy) models in a UML or Entity Relationship like model. A question which results from this is, when using models, is it possible to automatically generate a part of business rules? For doing so by machines, they need to understand formally (semantics) terms and concepts they are manipulating.

In Model Driven Architecture (MDA) every concept is expressed by a model, but it does not say anything about semantics. In another side, researches in Semantic Web, especially the use of ontologies, give many possibilities for adding semantics to semi-structured data, making automatic reasoning possible.

In this paper, we focus in “how can business rules be automatically generated from conceptual models semantically

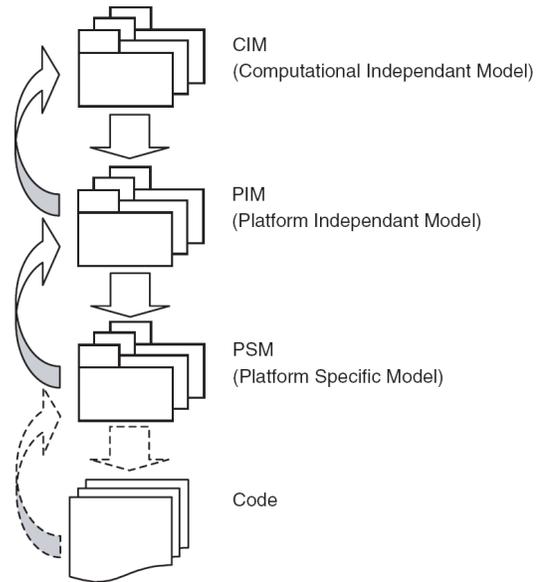


Fig. 1. Global view of the Model Driven Architecture approach

enriched? And what is the interest in doing so?”

This paper will first discuss Model Driven Architecture and semantics. We will present potential solutions for enriching MDA models with semantics. At last, we will also discuss the possibilities and the benefits provided by mixing models and web reasoning.

II. MODEL DRIVEN ARCHITECTURE

The Model-Driven Architecture starts with the well-known and long established idea of separating the specification of the operation of systems from the details of the way that these ones use the capabilities of their platform. Figure 1 gives a general view of the MDA approach. We can see that a construction of a new Information System begins with the development of one or more requirement models called Computation Independent Model (CIM). Then we may develop models independent from any platform called Platform Independent Model (PIM). In theory, the latter models must be partially generated from the

former. 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 informations 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 do exist.

A. MDA models and semantics

MDA principals are very interesting and allow economizing time 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. So adding semantics will offer a more interesting way for automatic generation.

1) *Why should MDA take care about semantics:* Making transformations between CIM and PIM, between PIM and PSM, and between PSM and code are done by specifying transformation rules. Nowadays these rules are handwritten and machines cannot generate them 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 part of them.

2) *Potential solutions for adding semantics to models:* In MDA, an instance of MOF (Meta Object Facilities) is used for representing models but our works are only concerned by UML models. For adding semantics to UML models we can use:

- 1) *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 an object application, a metamodel, a model, a database structure or anything else just by looking at it. For adding precision, the OMG has standardized the concept of UML profile. A UML profile is a set of technics and mechanisms allowing to adapt UML to a particular and specific domain. UML profile can be used in any UML models and does not modify the structure of the metamodel. UML profiles are stereotypes or labels which can be pasted on models. After having pasted labels on models, we can make inference using them. As we can see, doing this can solve our problem of semantics lack on models 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.
- 2) *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 this purpose. OCL allows expressing many kinds of constraints on UML models. For example, we can express constraints like: “before renting a car to a person one must be sure

that this person is ok”. OCL seems to be a good solution for our problem but it is not the case. Indeed, the first problem with OCL is that it does not offer automatic inference for machines and the second is that it does not support side effect operations. However OCL 2.0 does permit reference to operations that change the state of the system in a constraint expression, but the semantics of such a reference is that the operation will have been invoked when the truth of the constraint is tested. This semantics, which is permitted only in post-conditions, does not satisfy the requirements of the action clause of production rules, which cannot be used as postconditions of operations.

- 3) *Action Semantics:* remember that the main constraint with OCL was that it only supports no side effects operations. To solve this constraint, the OMG standardized Action Semantics [10]. Now we have a formalism which is able to express any kind of operations and constraints but it is not enough. Indeed, this formalism is too complicated to be used, 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 “technics” proposed so far is suitable for our purpose, which we recall, consists in exploiting semantics by machines.

In another side a new domain of computer science is growing more and more: Semantic Web. The aim of the Semantic Web is to make the web comprehensible by both humans and machines. A part of Semantic Web is about ontology and reasoning. Modeling concepts 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)[24] supports the inclusion of certain types of constraint in ontologies, allowing new information to be deduced when combining instance data with these description logics.

B. The Ontology Definition Metamodel

The MDA and its four-layer architecture provides a solid basis for defining the metamodels of any modeling language, and thus a language for modeling ontologies based on the MOF [2]. ODM is a proposal for an OMG’s RFP (Request For Proposal) resulting from an extensive previous research in the fields of the MDA and ontologies [4], [5], [8]. The main goal of ODM is to bridge the gap between traditional software tools for modeling (like UML) and artificial intelligence technics (Description Logics) for making ontologies. The principle of ODM is to merge two big domains of research which are Model Driven Architecture and Semantic Web. ODM is still in standardization process at the OMG [7] when this paper was being written. Basically the ODM allows making ontologies using UML (by using an UML profile with existing tools like Rational Rose or Poseidon) and transforming it to OWL/RDF, Topic Map or Common Logic (Figure 2).

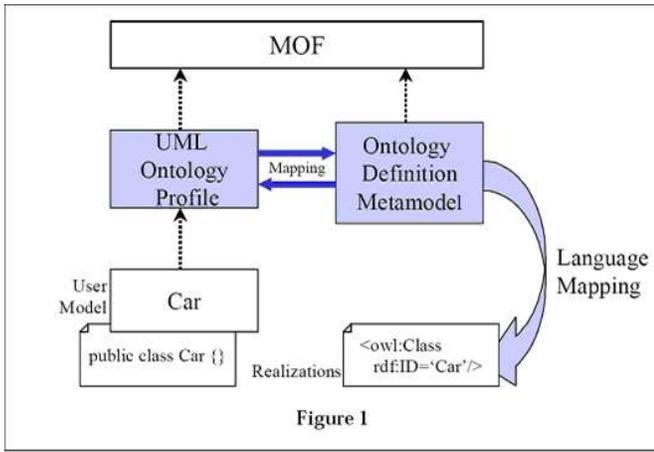


Fig. 2. ODM principle

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

III. ONTOLOGY REASONING

Ontology is an area of great importance for the semantic web. An ontology establishes the things that a system can talk about and makes reasoning on. Describing concepts and relationships between them formally, offers to machines the ability of making some varieties of logic, formally or not. Ontology supplies the concepts and terms; logic provides ways to make statements that define and use them, and to reason about collections of statements that use the concepts and terms. In the semantic web, logic plays many different roles:

- 1) Firing rules: having a set of facts, take a decision.
- 2) Making inference on facts.
- 3) Explaining why a particular decision has been reached.
- 4) Detecting contradictory statements and claims.

OWL exploits results of more than 15 years of Description Logics (DL) research. Indeed, for OWL a semantics was defined such that very large fragments of the language can be directly expressed using so called Description Logics. Description Logics is a family of logic based Knowledge Representation formalisms descendants of semantic networks and it describes domains in terms of concepts (classes), roles (properties, relationships) and individuals. In description logics terminology, a tuple of a T-box and an A-box is referred to as a knowledge base. An individual is a specific named object. With some restrictions, one can state that the logical basis of OWL can be characterized with the description logics $SHIQ(D_n)^-$ [?]. This means, with some restrictions, OWL documents can be automatically translated to $SHIQ(D_n)^-$ T-boxes. The RDF-Part of OWL documents can be translated to $SHIQ(D_n)^-$ A-boxes.

The logic $SHIQ(D_n)^-$ is interesting for practical applications because highly optimized inference systems are available (e.g. Racer). Given the background of Description Logic, these inference services can be used to solve actual problems with OWL knowledge bases. Reasonings are done using both

domain and range restrictions, individuals and also properties' characteristics (functional, transitivity, symmetric, inverses, etc.)

The Figure 3 describes our approach using ODM, our

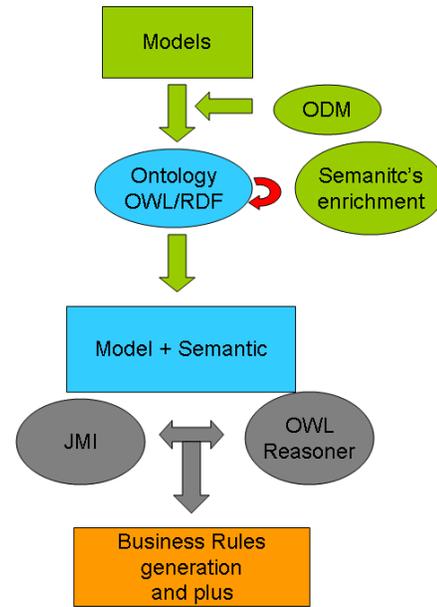


Fig. 3. Our approach

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 rules: serialize the rich model in XMI or use e.g JMI for parsing it manually.

IV. ADDING SEMANTICS TO MODELS FOR AUTOMATIC BUSINESS RULES GENERATION

MDA technologies and Semantic web are complementary; the former is concerned about automating the physical management and interchange of metadata, while the latter is focused on the semantics embodied in the content of the metadata as well as on automated reasoning over that content. The Semantic Web is the new-generation Web that tries to represent information such that it can be used by machines not just for display purposes, but also for automation, integration, and reuse across applications. Model Driven Development (MDD) is being developed in parallel with the Semantic Web. Emerging applications in finance, healthcare, security, communications, business intelligence, and many other vertical markets are content and context sensitive (semantics), and require enterprise scalability and performance. Merging Semantic Web and MDA technologies can fill this lack. Merging these two domains will be beneficial to both:

- 1) MDA is only interested by content and not by context (semantics), semantic web will resolve this important problem.
- 2) For semantic web: an interesting thing is that so mature UML tools could be used for making ontologies rather than using so theoretical languages from Artificial

Intelligence domain. In companies software engineers usually are doing models with UML, so it will be a good thing for allowing them using their preferred UML tools for making Ontology. Doing so will facilitate the use of ontologies.

Merging MDA and Semantic Web technologies allows more automatic processing like generation of constraints and business rules from models.

A. Our Approach for business rules automatic generation

Our principle is to use the advanced researches in Semantic Web, to combine it with Model Driven Architecture in order to make automatic business rules generation.

For generating business rules automatically, we will use principally the semantics in OWL format. In OWL reasoning, we can make automatic reasoning both with structures (TBox) and assertions on individuals and properties (ABox). 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 rules: serialize the rich model in XMI and use e.g JMI for parsing it manually. Another solution is making inference directly with the OWL model using an OWL reasoner. We have adopted the last solution because there exist good OWL Reasoners and this solution uses less intermediary steps.

Recall that on goal is not to generate all kinds of business rules. Indeed, this is infeasible. However, the part of them that able to generate will save time for business experts. Figure 4 summarizes our approach throughout MDA layers. As we

Model using our rule language [11] and models based on XMI like standard. At the PIM level either our business rules language ERML, the RIF W3C standard, the PRR OMG proposal or RuleML [6] may be used. At this step we use our “translators” for generating rules at the PSM level for a specific rule engine.

V. CONCLUSION

A business rules application is intentionally built to accommodate continuous changes in business rules. The ability to change them effectively, is fundamental for improving business adaptability. The platform on which the application runs should support such continuous changes. Offering to knowledgeable business people (experts) the possibilities to formulate, validate, and manage rules in a “zero-development” environment brings more value-added to this notion of “computer sciences in humanity’s service”. Allowing an automatic generation part of this business rules will be of valuable help. In this paper, we have seen that, by combining the two domains, Model Driven Architecture and Semantic 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). Due to the fact that the standardization of SBVR is recent, no implementation does exist. The best to our knowledge, the only free implementation we know is SBeaVER which is not at an advanced level and, at this moment, can only express Business vocabulary and not Business Rules.

Adding semantics to conceptual models open exciting and interesting domains of applications like information merge.

REFERENCES

- [1] Ronald G. Ross. *Principles of the Business Rule Approach*. Addison-Wesley, Boston, USA, 2003.
- [2] Dragan Gaëvić, Dragan Djurić, and Vladan Devedžić. *Model Driven Architecture and Ontology Development*. Springer-Verlag, Berlin, DE, 2006.
- [3] Kuldar Taveter and Gerd Wagner. *Agent-Oriented Enterprise Modeling Based on Business Rules*. In Springer-Verlages, editor, *20th International Conference on Conceptual Modeling (ER2001)*, LNCS, November 2001.
- [4] 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 Semantic Web*. *Lecture Notes in Computer Science*, 2185:342+, 2001.
- [5] Dragan Djuric, Dragan Gasevic, and Vladan Devedzic. *Ontology Modeling and MDA*. *Journal of Object Technology*, 4(1):109–128, 2005.
- [6] RuleML. *The RuleML initiative*.
- [7] The Object Management Group OMG, IBM, and Sandpiper Software. *Ontology Definition Metamodel*. OMG Specification, June 2006.
- [8] Stephen CraneField. *Networked Knowledge representation and exchange using UML and RDF*. *Journal of digital information*, 1(8), 2001.
- [9] Elisa F. Kendall, Mark E. Dutra, and Deborah L. McGuinness. *Towards A Commercial Ontology Development Environment*. In *Proceedings of the 1st International Semantic Web Conference (Posters and Demos)*, 2002.
- [10] The Action Semantics Consortium. *Action semantics for the uml*. OMG Specification (ad/2001-03-01), March 2001.
- [11] Mouhamed Diouf, Kaninda Musumbu, and Sofian Maabout. *Standard Business Rules Language: why and how?* *The 2006 International Conference on Artificial Intelligence*, June 2006.

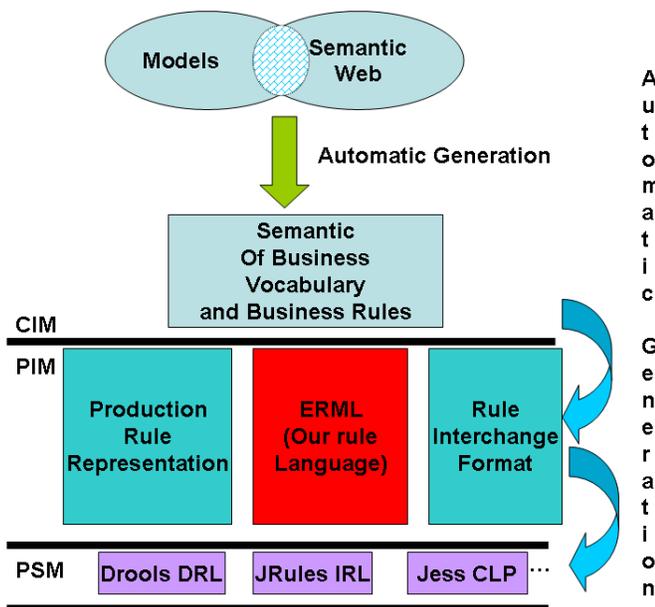


Fig. 4. Our approach throughout the MDA layers

can see the first step will be a generation according to the Computation Independent Model (CIM) in an OMG SBVR like syntax (in strict natural language), the next step will be to generate executive rules according to the Platform Independent