Generating and Merging Business Rules by Weaving MDA and Semantic Web

Mouhamed Diouf, Sofian Maabout and Kaninda Musumbu

Université Bordeaux I, LaBRI (UMR 5800 du CNRS), Domaine Universitaire 351, cours de la Libration 33405 Talence Cedex, France

Abstract. Information systems (IS) are getting more and more complex. The design of such systems requires various individuals with varied expertises. The requirements expressed, when the design of such a system is decided, may change upon time. These changes may even occur very often. Thus, the more the system is flexible, the easier the upgrades will be. One of the standard ways to design flexible systems is the use of the so called “business rules” whose aim is the separation of business from system in an application. Business rules define and constrain business processes in enterprizes. 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 way to automatically generate and merge a part of the business rules by combining Model Driven Architecture and the Semantic Web using the Ontology Definition Metamodel.

1 Introduction

Business rules are statements that express (certain parts of) a business policy, defining terms and defining or constraining the operations of an enterprize, in a declarative manner [1–4]. The business rule approach is more and more used due to the fact that in such system, business experts can maintain the complex behavior of their application in a “zero development” environment. There exist more and more business rule management systems (BRMS) and rule engines, adding new needs in the business rules community. Currently the main need in this domain is having a standard language for representing business rules, facilitating their integrations and share. Work for solving this lack is in progress at OMG and W3C [5–9] and others initiatives [10, 11].
In another side, an enough heavy step during business rules bases systems realization is the step of elicitation of rules from the business. Entreprises, generally, have (legacy) models in a UML or Entity Relation like model. A question which results from this is, using models, is it possible to automatically generate a part of business rules? For doing automatic generation by machines, they need to understand formally (semantics) terms and concepts they are manipulating.
In Model Driven Architecture (MDA) [12] every concept is expressed by a model, but it does say anything about semantics [13]. In another side, researches in Semantic Web, especially the use of ontologies, give many possibilities for adding semantics to semistructured data, making automatic reasoning possible. Logic is a key component

Diouf M., Maabout S. and Musumbu K. (2007).
Generating and Merging Business Rules by Weaving MDA and Semantic Web.
In *Proceedings of the 3rd International Workshop on Model-Driven Enterprise Information Systems*, pages 19-28
DOI: 10.5220/0002427000190028

in the Semantic Web. Also, the Semantic Web will be one of the biggest open systems, so merging information coming from different sources with different reliability level, in a correct way will be crucial. In this paper, we focus in "how can business rules be automatically generated and merged from conceptual models semantically enriched? And what is the interest in doing so?"
The paper is divided in two main parts. Section 1 is devoted to an introduction to business rules and the concept of Model Driven Architecture and semantics. In section 2, we give a methodological overview of our approach and discuss the possibilities and the benefits provided by mixing models and web semantics. We insist of its wide applicability and theoretical soundness.

2 The Business Rules Approach

2.1 Business Rule's Definition

The aim of business rules is to capture customer needs [14]. The term business rule has a different meaning according to business or system point of view:

- From an information system perspective: "A business rule is a statement that defines or constrains some aspect of the business (policies, know-how). It is intended to assert business structure, or to control or influence the behavior of the business." [3].
- From a business perspective: "A business rule is a directive, which is intended to influence or guide business behavior, in support of business policy that is formulated in response to an opportunity or threat." [15].

In a more simple and basic formulation, a business rule is a pair of if-then statements (we will see it in more details in next sections).

Principles and Objectives. In information systems, we do no more ask "whether the behavior will change" but rather "when the behavior will change". So it is a good thing to handle those changes from the beginning of the conception step. Business rules are at the center of any application, however hardest is to list them and to structure them in an effective approach of management [1]. The use of a Business Rules Management System (BRMS) facilitates the census and the implementation of business rules [1].
The business rules approach makes it possible to allow a "collaboration" place between business and system experts. It allows legacy code curation that becomes through time very difficult and heavy to maintain and also to make evolve.
The principle of business rules is not a new one, it is a direct application of the Seventies artificial intelligence theories [16] which, at that time, passed to be utopian due to a problem of computer's power.
The objective of the business rules approach is to allow information systems design guided by the business, for the business and by the business [3]. To the traditional criteria of information systems, namely robustness, maintainability, extensibility, scalability, a new one is added: adaptability [14]. A system must be able to be adapted to the fluctuations of the market. Further more, the behavior of such systems must be directly

modified by business people in a zero-development environment. Who know the business of the application better than them? With this, requests of update will no longer inundate the information technology service with work. Business maintenance will no longer wait eternal time to be processed. The business rule approach allows satisfying these requirements by separating the business logic (the what) and the system logic (the how). The business rules approach makes it possible for business experts to manage the behavior of the system using natural languages (business languages).
The hardest task in making business rules based systems is to extract them from knowledge. Generally, knowledge is at the end represented by models. In the next sections, we will talk about MDA models and show how can we use them for extracting business rules.

3 Model Driven Architecture

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 a system uses the capabilities of its platform [12].

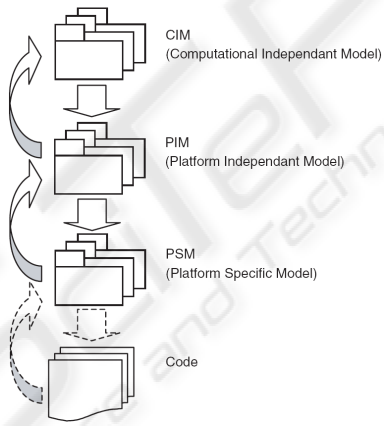

Fig. 1. Global view of the Model Driven Architecture approach.

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 requirements models (CIM). Then we may develop models independent from any platform (PIM). In theory, the latter models must be partially generated from the former. PIM 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) that are obtained by transforming PIM and by 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 do exist.

3.1 MDA Models and Semantics

MDA principles 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 using semantics will offer a more interesting way in automatic generation.

Business rules are about meanings and act on models. Generating all business rules is impossible but it would be possible and helpful to generate a large part of them. For doing this automatically, it is clear that adding semantics in models is needed.

Some Solutions for Adding Semantics to Models. In MDA, an instance of MOF (Meta Object Facilities) [17] is used for representing models but our work is 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 [13]. For adding precision, the OMG has standardized the concept of UML profile [18]. A UML profile is a set of technics allowing to adapt UML to a particular domain. Once these profiles are pasted on models, we can use them for making inference. 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 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 [19] was standardized by OMG for this purpose. OCL allows expressing many kinds of constraints on UML models, e.g "before renting a car 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 is that it does not offer automatic inference 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 post-conditions 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 [20]. 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 [13], was not created while thinking to machine comprehension and self-use, and does not have a textual formalism.

As we can see, none of the UML "technics" is suitable for adding semantics to models. In another side the semantic web aims to make the web comprehensible by both humans and machines [21]. A part of semantic web is about ontology and reasoning. Ontologies are used to model concepts, relationships between them, properties and instances of those concepts [22]. In addition, the Web Ontology Language [23] supports the inclusion of certain types of constraints in ontology, allowing new information to be deduced when combining instance data with these description logics [22]. At this point our dilemma is how can we use MDA models and Semantic Web? Ontology Definition Meta Model (ODM) is the response.

3.2 The Ontology Definition Metamodel

MDA provides a solid basis for defining the metamodels of any modeling language, and thus a language for modeling ontologies based on the MOF [24]. The ODM is a proposal for an OMG's RFP (Request For Proposal) [25] resulting of an extensive previous research in the fields of the MDA and ontologies [26–28]. 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 Model Driven Architecture and Semantic Web. ODM is still in standardization process at the OMG [29] when this paper is being written. Basically ODM allows creating ontologies using UML and transforming it to OWL/RDF, Topic Map or Common Logic (Figure 2).

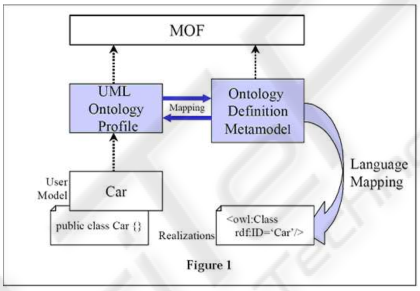

Fig. 2. ODM principle.

Next section, we see how ontology reasoning is used to solve the lack of semantics in models.

4 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 [30]. Model Driven Development (MDD) is being developed in parallel with the Semantic Web [31]. Emerging applications in finance, healthcare, security, communications, business intelligence, and many other vertical markets are content and context sensitive (semantics) [30]. Merging Semantic Web and MDA will be benefic to both:

– MDA is only interested by content and not by context (semantics), semantic web will solve this important problem.
– For semantic web: an interesting thing is that so mature UML tools could be used for making ontologies rather than using so theoretical languages. Very often, software engineers use UML, so it will be a good thing for allowing them using their preferred UML tools for modeling Ontologies. Doing so will facilitate the use of ontologies.

Merging MDA and Semantic Web technologies allow more automatic processing like: generation of constraints and business rules from models. Now for example, suppose that, to our little model in Figure 3 we add an ontology where we declare that a human must have a mother that must be a human to. Therefore, with qualified "reasoners" we can generate that: IF a *Human* is the mother of a *Human* then that *Human* is a *Woman*. Therefore, we can infer that *"IF Christ mother's name is Marie THEN Marie is a Woman"*.

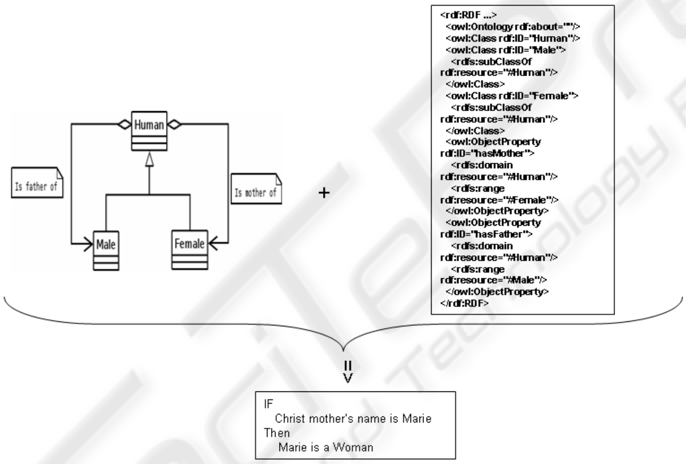

Fig. 3. A little ontology for a little model.

4.1 Our Approach for Business Rules Automatic Generation

For generating business rules automatically, we will use principally the semantics in OWL format. In OWL, we can make automatic reasoning with both structures (TBox) or assertion on individuals and properties (ABox) [32]. In our case for example, if we have: $Predicate : Domain1 \longmapsto Domain2$. This declaration means that we have a property *Predicate* going from the domain *Domain1* to the range *Domain2*. So we want to generate that:

$$\begin{array}{ll} IF & Object1\ Predicate\ Object2 \\ THEN & Object1\ is\ of\ type\ Domain1 \quad AND \quad Object2\ is\ of\ type\ Domain2 \end{array}$$

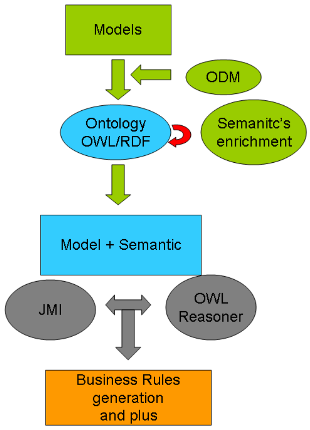

Fig. 4. Our approach.

Figure 4 describes our approach: using ODM, our model is generated in OWL/RDF model and this last one is enriched with semantics. Now two solutions are possible for generating rules: serialize the rich model in XMI [33] and use e.g JMI [2] for parsing it manually or making inference directly with the OWL model using a OWL reasoner. We have adopted the last solution because there exist good OWL Reasoners and this solution uses less intermediary steps.

Recall that our gaol is not to generate all kinds of business rules. Indeed, this is infeasible. However, the part of them that we will be able to generate will save time for business experts. Figure 5 summarizes our approach throughout MDA layers.

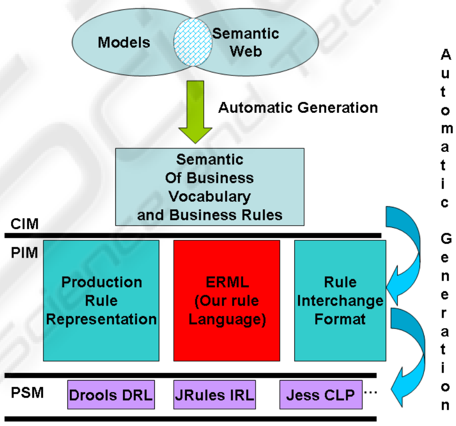

Fig. 5. Our approach throughout the MDA layers.

As we can see the first step is a generation according to the CIM in an OMG SBVR [5] like syntax (in strict natural language), the next step is to generate executive rules according to the PIM and models based on XMI like standard. At the PIM level either our business rules language ERML [34], the RIF W3C standard, the PRR OMG proposal or RuleML [6, 7, 11] may be used. At this step we use our "translators" for generating rules at the PSM level for a specific rule engine. We are still missing a standardized rule language.

5 Merging and Aligning Business Rules

Information sharing is the key issue in cooperative information systems. One particular advantage of sharing information among multiple systems is that it is often able to deduce additional knowledge that is not locally held by any of the systems, but collectively by all of them. However, information from different systems might, and often do, conflict each other. Logic is a key concept in the Semantic Web and will come from different entities and sources. Business rules coming from different sources with different reliability level and intervening at the logical level in the Semantic Web stack, must be merged or aligned. This work must be done regardless of whether the ultimate goal is to create a single coherent rule base that includes the information from all the sources (**merging**) or if the sources must be made consistent and coherent with one another but kept separately (**alignment**) [35]. The issue of modularity in logic programming has been, during the 90s [36], and is still, actively investigated. For example, suppose that we have a model M_A with a data base DB_A in which we have tuples $Name, Birthtown$ and a set of rules KB_A formed by:

- R_{A_1}: *IF* Name is empty *THEN* throw error
- R_{A_2}: *IF* Birth town is empty *THEN* throw error

Moreover suppose we have a model M_B with a data base DB_B in which we have tuples $Name, Basetown$ and a set of rules KB_B formed by:

- R_{B_1}: *IF* Name is empty *THEN* cancel
- R_{B_2}: *IF* Base town is empty *THEN* cancel

If we want to use both M_A and M_B in the same system, we will probably want to use DB_A and DB_B and merge KB_A and KB_B. Doing so, a conflict is raised: R_{A_1} and R_{B_1} have the same conditions but actions are not the same. How to do? Currently the work of merging, or aligning business rules is performed mostly by hand, without any tool to automate the process fully or partially. The experience of manually merging and aligning the knowledge bases is an extremely tedious and time-consuming process. At the same time we can notice that many steps in the process could be automated, many points where a tool could make reasonable suggestions, and many conflicts and constraints violations for which a tool could help.

Our approach presented in previous sections can be used for merging knowledge base in a semi-automatic way based on a semantic level rather than syntactic. Like business rules generation, first models transformed in ontologies are enriched semantically. Then we use the PROMPT [35] Algorithm for merging ontologies. PROMPT takes two ontologies as input and guides the user in the creation of one merged ontology as output. All terms and concepts used in business rules constitute the ontologies, so merging business rules is equivalent to merge ontologies. We choose PROMPT because, due to his extremely general knowledge model (Classes, Slots, Facets and Instances), it can be applied over a variety of knowledge representation systems. A part of business rules could be automatically merged (like R_{A_2} and R_{B_2}), and the other part will be presented to business experts for validation (like R_{A_1} and R_{B_1}).

6 Conclusion

A business rules application is intentionally built to accommodate continuous change 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 change. Offering to knowledgeable business people (experts) the possibility 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". Automatically generating a part of these business rules will be valuable. In this paper, we have seen that by combining Model Driven Architecture and Semantic Web, a solution is possible. Right now we can only make generation according to the CIM in a OMG SBVR like syntax. The next step will be to generate executable rules according to the PIM using our rule language or models based on XMI. The last step will be to have an editor allowing to edit both models and semantics.
Manually merging and aligning knowledge bases is an extremely tedious process. We also have seen that using MDA and Semantic Web a semi-automatic solution is possible for merging and aligning rule bases. Making simple generic business rules generation from models possible will facilitate the use of the business rules approach.
Adding semantics on conceptual models will open an exciting and interesting domain of application because, ontologies reasoning principle becomes possible with MDA models.

References

1. Barbara von Halle. *Business Rules Applied*. John Wiley & Sons, New York, USA, 2002.
2. Java Community Process(JCP). *Java Metadata Interface (JMI)*. Sun (JSR 40), 2002.
3. Ronald G. Ross. *Principles of the Business Rule Approach*. Addison-Wesley, 2003.
4. Kuldar Taveter and Gerd Wagner. *Agent-Oriented Enterprise Modeling Based on Business Rules*. In Springer-Verlags, editor, *20th International Conference on Conceptual Modeling (ER2001)*, LNCS, November 2001.
5. The Object Management Group OMG. *Semantics of Business Vocabulary and Business Rules (SBVR)*. OMG Specification, March 2006.
6. W3C. *Rule Interchange Format (RIF)*. W3C Workgroup, 2005.
7. The OMG. *Production Rule Representation (PRR) RFP*. OMG Request For Proposal (br/2003-09-03), 2003.
8. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. *W3C Member Submission*, May 2004.
9. W3C. *Rule interchange format Workgroup, http://www.w3.org/2005/rules/*, 2005.
10. IBM T.J. Watson Research Center. *CommonRules project*. Intelligent Agents project 1997.
11. RuleML. *The RuleML initiative*.
12. The Object Management Group OMG. *Model Driven Archtecture Guide Version 1.0.1*. OMG Specification, June 2003.

13. Xavier Blanc. *MDA en action*. Eyrolles, France, 2005.
14. Ronald G. Ross. *The Business Rule Book*. Business Rule Solutions, 1997.
15. D. Hay and K. A Healy. *GUIDE Business Rule Project Final Report*. Technical report, October 1997.
16. Ernest Friedman-Hill. *JESS in Action*. Manning Publications Co, Greenwich, UK, 2003.
17. The Object Management Group OMG. *Meta Object Facility (MOF)1.4*. OMG Specification (formal/02-04-03), April 2002.
18. The Object Management Group. *Unified Modeling Language: Superstructure*. OMG Specification, February 2004.
19. The Object Management Group OMG. *UML 2.0 OCL Specification*. OMG Specification, October 2003.
20. The Action Semantics Consortium. *Action semantics for the uml*. OMG Specification (ad/2001-03-01), March 2001.
21. Thomas B. Passin. *Explorer's guide to the Semantic Web*. Manning Publications Co, Greenwich, UK, 2004.
22. Stephen Cranefield and Jin Pan. *Bridging the Gap Between the Model-Driven Architecture and Ontology Engineering*. *Proc. of AOSE 2004 Workshop*, 2004.
23. W3C OWL M. K. Smith, C. Welty, and D. L. McGuinness. *OWL Web Ontology Language Reference*. W3C Standard, February 2004.
24. Dragan Gaěvič, Dragan Djurié, and Vladan Devedžić. *Model Driven Architecture and Ontology Development*. Springer-Verlag, Berlin, DE, 2006.
25. The Object Management Group OMG. *Request For Proposal for Ontology Definition Metamodel*. OMG Request For Proposal, March 2003.
26. 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.
27. Saartje Brockmans, Raphael Volz, Andreas Eberhart, and Peter Löffler. *Visual Modeling of OWL DL Ontologies Using UML*. In *International Semantic Web Conference*, pages 198–213, 2004.
28. Dragan Djuric, Dragan Gasevic, and Vladan Devedzic. *Ontology Modeling and MDA*. *Journal of Object Technology*, 4(1):109–128, 2005.
29. The Object Management Group OMG, IBM, and Sandpiper Software. *Ontology Definition Metamodel*. OMG Specification, June 2006.
30. H. Knublauch. *Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protege/OWL*. *1st International Workshop on the Model-Driven Semantic Web (MDSW2004)*, 2004.
31. Stephen J. Mellor, Anthony N. Clark, and Takao Futagami. *Guest Editors' Introduction: Model-Driven Development*. *IEEE Software*, 20(5):14–18, 2003.
32. Raphael Volz. *Web Ontology Reasoning with Logic Databases*. PhD thesis, Universität Karlsruhe (TH), Universität Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe, 2004.
33. The Object Management Group OMG. *MOF 2.0/XMI Mapping Specification, v2.1*. OMG Specification (formal/05-09-01), 2005.
34. Mouhamed Diouf, Kaninda Musumbu, and Sofian Maabout. *Standard Business Rules Language: why and how?* *The 2006 International Conference on Artificial Intelligence*, June 2006.
35. Natalya Fridman Noy and Mark A. Musen. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. In *AAAI/IAAI*, pages 450–455, 2000.
36. Michele Bugliesi, Evelina Lamma, and Paola Mello. *Modularity in Logic Programming*. *J. Log. Program.*, 19/20:443–502, 1994.