Transformation of SBVR Compliant Business Rules to Executable FCL Rules

Aqueo Kamada[1,4], Guido Governatori [2] and Shazia Sadiq[3]

[1] CTI, Rod. Dom Pedro I, km 143.6, Campinas, Brazil
[2] NICTA, Queensland Research Laboratory, Brisbane, Australia
[3] The University of Queensland, ITEE, Brisbane, Australia
[4] Unicamp, FT, Limeira, Brazil
aqueo.kamada@cti.gov.br,
guido.governatori@nicta.com.au,
shazia@itee.uq.edu.au,

Abstract. The main source of changing requirements of the dynamic business environment is response to changes in regulations and contracts towards which businesses are obligated to comply. At the same time, many organizations have their business processes specified independently of their business obligations (which include adherence to contracts laws and regulations). Thus, the problem of mapping business changes into computational systems becomes much more complicated. In this paper we address the problem by providing an automated transformation of business rules into a formal language capable of directly mapping onto executable specifications. The model transformation is consistent with MDA/MOF/QVT concepts using ATL to perform the mapping. Business rules are compliant to SBVR metamodel, and are transformed into FCL, a logic based formalism, known to have a direct mapping onto executable specifications. Both, source and target rules are based on principles of deontic logic, the core of which are obligations, permissions and prohibitions.

Keywords: Business Contract; Business Rule Transformation; SBVR; FCL; MDA

1 Introduction

Due to the current dynamic and highly competitive business environment the organizations have to make changes in their computational systems in a much more accelerated rhythm than in past decades. Consequently, the computational solutions for the business problems cannot accompany the speed in which the change necessities appear. One of the main sources of change is response to changes in regulations and contracts towards which businesses are obligated to comply.

Commonly, documents containing contracts, regulations, laws and procedures define the strategies, policies and relationships among organizations and consolidate the organization´s knowledge. From those documents arise the rules that define the behavior of the business processes in the organizations [1]. Hence, the computational systems must be compliant with these business documents. So, ensuring compliance

M. Dean et al (eds)
RuleML 2010, The 4th International Web Rule Symposium
LNCS 6403, pp. 151-159

of business processes with business contracts means ensuring consistency of rules stated in business contracts and rules covering the execution of business processes.

We propose an MDA (Model Driven Architecture) [2] based model to transform SBVR compliant business rules [3] extracted from business contract of services to compliant executable rules in FCL – Formal Contract Logic [4]. Both business rules and FCL rules are based on principles of deontic logic [5] for treating expressions in the form of normative policies, the core of which are obligations, permissions and prohibitions. Deontic constraints express what parties to the contract are required to perform (obligations), what they are allowed to do (permissions), or what they are not allowed to do (prohibitions). We also present a transformation exercise using ATL (Atlas Transformation Language) [6] to transform SBVR compliant rules to FCL rules. We do not go into details about the generation of the predicates of FCL.

The next section provides an overview of the MDA modeling framework and Section 3 discusses aspects on business rules and business contracts formalization. Section 4 presents some requirements on the business contracts edition. Section 5 presents the proposed model transformation and Section 6 discusses some related works and the final section provides a conclusion and discussion on future researches.

2 Foundations on Model Transformation

Model transformation is the process of transforming a model, say Ma, conforming to metamodel MMa into a model, say Mb, conforming to metamodel MMb. QVT (Query/View/Transformation) [7], is an OMG (Object Management Group) standard for performing model transformations in the context of MDA and it can be used to do syntactic or semantic transformation.

The idea of Model Driven Engineering is that, through transformations accomplished on the conceptual model, new models are generated, with abstraction levels more and more specific and the final system is generated automatically. The built models are formals, avoiding ambiguity, so that they can be understood by software systems.

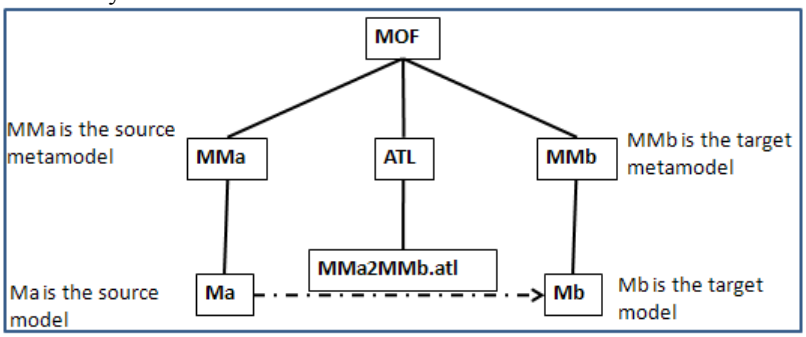

Figure 1. ATL transformation context.

ATL is a model transformation language developed by OBEO and INRIA to answer the QVT Request For Proposal. Considering the Figure 1, an ATL program (MMa2MMb.atl) will take model Ma.xmi as input and will produce model Mb.xmi as

output. Both models may be expressed in the OMG XMI [8] standard. The model Ma conforms to metamodel MMa.km3. Model Mb conforms to metamodel MMb.km3. The KM3 (Kernel MetaMetaModel) notation is a simple and neutral metamodel specification language. The ATL program itself (MMa2MMb.atl here) is also a model, so it conforms to a metamodel (the ATL metamodel) not presented here. An ATL program is composed of a header, of a set of side-effect free functions called helpers and of a set of rules.

3 Formalization of Business Rules and Business Contracts

This section presents some foundations on business rules and business contracts and discuss some aspects related to their formalization.

3.1 Business Rules

Although there are a lot of discussion around the definition of what "business rule" means [3], [9] in the context of this work, a business rule is "a rule that can be interpreted by computers, that defines or restricts some aspects of a business, introducing obligations or needs, according to the organization policies."[10]. Following are some business rules in the context of car rental:

- A car must have a registration number.
- A car should not be released to the customer if the credit card was not presented as the payment guarantee.
- A driver of a rental car must be a qualified driver.

The main objective of the SBVR metamodel [3] is to allow business people to define the policies and the rules that drive the organizations in the business people's own language, in terms of the artifacts with which they perform the businesses. Besides, the other objective is to capture those rules in a clear way, without ambiguity, and quickly transformable in other representations, as the representations for business people, for software engineers, and for business rules execution tools.

According to SBVR metamodel a business rule can be expressed formally in statements in a structured English language using a font style convention. There are four font styles with formal meaning: (i) term - the 'term' font is used for a designation for a noun concept (other than an individual concept); (ii) Name - the 'name' font is used for a designation of an individual concept that tend to be proper nouns (e.g., Washington); (iii) *verb* - the 'verb' font is used for designations for fact types — usually a *verb*, preposition, or combination thereof; and (iv) `keyword` - the 'keyword' font is used for linguistic symbols used to construct statements – the words that can be combined with other designations to form statements and definitions (e.g., '**each'** and '**it is obligatory that'**). For example, in the business rule, a**s** shown in the Figure 2, includes three **keywords** or phrases, two designations for noun concepts and one for a *fact type*.

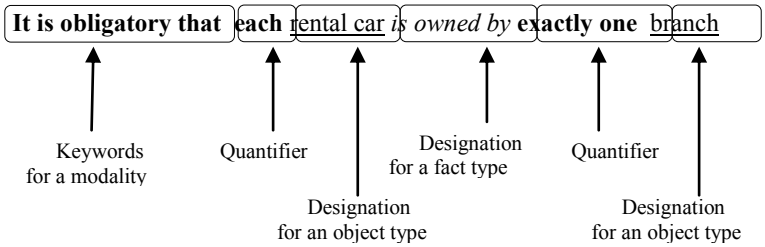

Figure 2. Business rule elements.

3.2 Business Contract of Services

This Section provides some issues related to contract of services formalization based on Formal Contract Logic (FCL). FCL was introduced in [11] for the formal analysis of business contracts and it is based on previous work on formal representation of contracts [12], logic of violations [13], and normative positions based on Deontic Logic with Directed Obligations [14].

A contract is structured in terms of a number of clause groups, each of which contains contract conditions. To save space, consider the following small part of the contract presented in [11] that will be analyzed and formalized in the subsequent sections.

CONTRACT OF SERVICES

This Deed of Agreement is entered into as of the Effective Data identified below.
BETWEEN *ABC Company (To be known as the Purchaser)*
AND *ISP Plus (To be known as the Supplier)*
WHEREAS *(Purchaser) desires to enter into an agreement to purchase from (Supplier) Application Server (To be known as (Service) in this Agreement).*
NOW IT IS HEREBY AGREED *that (Supplier) and (Purchaser) shall enter into an agreement subject to the following terms and conditions:*

…

5 Service Delivery

5.1 The (Supplier) shall ensure that the (Services) are available to the (Purchaser) under Quality of Service Agreement (http://supplier/qos1.htm). (Services) that do not conform to the Quality of Service Agreement shall be replaced by the (Supplier) within 3 days from the notification by the (Purchaser), otherwise the (Supplier) shall refund the (Purchaser) and pay the (Purchaser) a penalty of $1000.

5.2 The (Supplier) shall on receipt of a purchase order for (Services) make them available within 1 days.

…

Usually a contract comprises two types of clauses: definitional clauses giving the meaning of the terms used in the contract and clauses specifying the normative behaviors (i.e., giving the obligations, permissions, prohibitions the signing parties of the contract are subject to). We will concentrate only on the normative specifications

of a contract. Hence, we will ignore all the sections of the contract, except for the section 5. According to the normalization process in FCL [4] give us the following rules:

r5:1 : Service ⊢ O_SQualityOfService ⊗ O_SReplace3days ⊗ O_SRefund&Penalty ⊗ P_PChargeSupplier

r5:2 : PurchaseOrder ⊢ O_SDeliver1day ⊗ P_PChargeSupplier

4 Contract of Services Editor Requirements

As in any community, the users of the buyer and seller community use a common terminology, sharing the same understanding about the words, procedures and activities that are part of their daily business routine. To facilitate the task of business contract elaboration, Figure 3 gives an idea of how could be the external interface of an IDE – Integrated Development Environment. It should provide some editors and functionalities to define terms, facts, business rules, contracts and services using that community terminology.

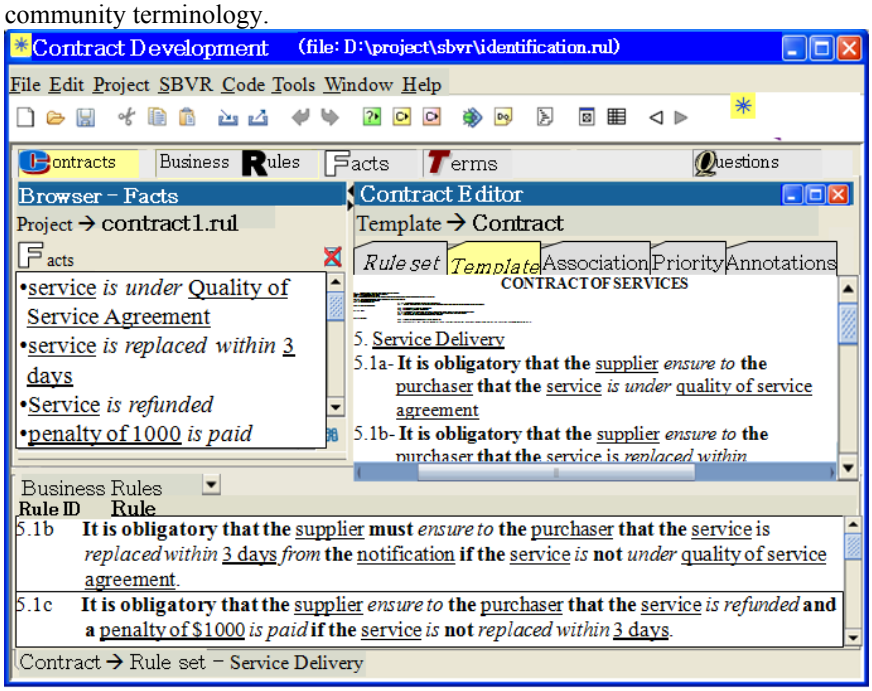

Figure 3. Interface Prototype for the Business Contract and Business Rules IDE

In this way, supplier, purchaser and service are terms designating concepts, which ultimately represent object types; "service *is replaced within* 3 days" and "service *is under* Quality of Service Agreement" are facts; and "**It is obligatory that the** supplier *ensure to* **the** purchaser **that the** service *is refunded* **and a** penalty of $1000 *is paid* **if the** service *is* **not** *replaced within* 3 days" is a business rule. All these elements are meaningful to that community and should be defined using the IDE.

When the business analyst defines terms, verbs, facts and rules contained in a contract of service, they will be stored in the IDE infrastructure.

5 Transformation of SBVR compliant rules to FCL rules

This Section presents the Contract of Services adherent to the SBVR metamodel and provides an example to exercise the proposed model transformation using ATL infrastructure to transform SBVR compliant elements to the FCL elements.

The prior Contract of Services could be represented by using the SBVR metamodel elements, i.e., in terms of terms (designation for object type), Names (designation for Name type), *verbs* (designation for fact type) and **keywords**. To save space, we will concentrate only on the normative specifications of the section 5.1 of the contract, which is divided into 3 business rules to easy understanding.

CONTRACT OF SERVICES

…

5. Service Delivery

5.1 –

a. It is obligatory that the supplier must *ensure to* the purchaser that the service *is under* quality of service agreement (http://supplier/qos1.htm).

b. It is obligatory that the supplier must *ensure to* the purchaser that the service is *replaced within* 3 days *from* the notification if the service *is* not *under* quality of service agreement.

c. It is obligatory that the supplier *ensure to* the purchaser that the service *is refunded* and a penalty of $1000 *is paid* if the service *is* not *replaced within* 3 days.

…

5.2 Model Transformation

Considering just the previous business rules 5.1b and 5.1c they should be transformed to the following FCL rules:

$r_{5:1b}$: ┐ ServiceIsUnderQoSAgreement ⊢ O_SServiceIsReplacedWithin3days

$r_{5:1c}$: ┐ ReplaceServiceWithin3days ⊢ O_SSupplierRefundsService, O_SSupplierPaysPenaltyOf$100

According to MDA's perspective we have to define the models for these text fragments. Following is the corresponding model, expressed in XMI, for the SBVR business rules. This model will be the input of the transformation mechanism.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns ="Rules">
 <Rule ruleId="r51b">
  <keyword keywordLabel="It is obligatory that"/>
   <keyword kw=" the"/>
```

```
    <fact factType="service is replaced within 3 days"/>
    <condition factType="service is not under quality of service agreement "/>
  </Rule>
  <Rule ruleId="r51c">
   <keyword keywordLabel="It is obligatory that"/>
    <keyword kw="the"/>
    <fact factType="service is refunded and penalty of $1000 is paid "/>
    <condition factType="service is not replaced within 3 days"/>
  </Rule>
 </ xmi:XMI >
```

Following is the corresponding model, expressed in XMI, for the FCL rules. This model will be the output of the transformation mechanism.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns ="FCL Rules">
 <Rule ruleId="r51b">
  <ant_premise>
      <symbol negation="not" />
      <premise a_prem= "ServiceIsUnderQoSAgreement"/>
    </ant_premise>
    <conc_premise>
      <premise c_prem= OSServiceIsReplacedWithin3days"/>
    </conc_premise>
 </Rule>
 <Rule ruleId="r51c">
  <ant_premise>
      <symbol negation="not" />
      <premise a_prem= "ServiceIsReplacedWithin3days"/>
    </ant_premise>
    <conc_premise>
      <premise c_prem="OSServiceIsRefunded"/>
      <premise c_prem="OSPenaltyOf$100IsPaid"/>
    </conc_premise>
 </Rule>
</xmi:XMI >
```

According to MDA and ATL phylosophy, these two models have to conform to the respective source and target metamodels. Thus, in order to achieve the transformation, it is necessary to provide: (i) a source metamodel in KM3 ("SBVR Rules"), (ii) a target metamodel in KM3 ("FCL Rules"), and (iii) a transformation model in ATL ("SBVR2FCL"). When the ATL transformation is executed the source model (XMI model for SBVR rules) will be transformed into the target model (XMI model for the FCL rules).

6 Related Works

This section discusses some works related to the business contract execution. These works mention the absence of an appropriate treatment so that the business contract

clauses and rules can be mapped into executable rules in a collaborative and integrated way with business process mechanisms.

Kabilan [15] proposes an approach to combine contract workflow models with Business Process Modeling Notation (BPMN) models. Business process modelers may model the contract obligation fulfillment process as Contract Workflow Models (CWM) using BPMN diagrams. The weakness of this proposal is that it is not complete in terms normative propositions based on Deontic Logic, for example, it cannot capture all informational aspects and related concepts, like prohibitions.

SweetDeal [16] is a rule-based approach to representation of business contracts that enables software agents to create, evaluate, negotiate, and execute contracts with substantial automation and modularity. It builds upon the situated courteous logic programs knowledge representation in RuleML. It combines RuleML with ontologies (DAML+OIL) for a practical e-business application domain. Although it seems to be a good approach, SweetDeal did not show how to incorporate legal aspects of contracts into the approach.

The Edee architecture [17] provides a mechanism for explicitly and uniformly capturing business occurrences, and provisions of contracts, policies, and law. Edee is able to reason about the interactions between organizations and execute business procedures informed by the combined legal effects of the corresponding diverse rules. It deals with both conflict detection and resolution. The weakness of Edee is that it does not show how effectively the business contract issues are translated to the dynamic context of executable business processes.

7 Conclusions and Future work

The proposed MDA based model transformation makes innovative contributions compared to other initiatives in mapping business contracts to executable code. The model (i) helps business analysts in the definition of contracts and rules, using a language familiar to them, using the terms with which they accomplish their businesses; (ii) can define contracts and rules, using templates, and express them in computation independent models (CIM); (iii) both business rules and FCL rules are based on principles of deontic logic for treating expressions in the form of normative policies, the core of which are obligations, permissions and prohibitions.

Besides, as a proof of concept for the proposed model transformation it is specified some requirements for the IDE to elaborate and edit business contracts, business rules, facts and terms. All these assets should be transformed to computational code, for rules and contract of services, adherent to FCL rules model.

The results indicate that the concepts, ideas and proposed model transformation are promising. Besides business contracts and rules formalization technologies, services (SOA), repositories and ontologies, it seems that the complete solution for the mentioned problems includes the following list of topics that deserve future researches:

- Inclusion of a mechanism in the IDE to contemplate process composition modeling using, for instance, languages such as BPMN and that could make transformation to executable languages like WS-BPEL.

- Proposition of a mechanism to help the business analyst to link business rule actions to Web services. May be considering Web 2.0 application facilities, such as recommendation system linked to trust, preference and rated content to create highly trusted environment for business analyst to decide which Web service is the most adequate in a specific rule action.
- Development of a prototype implementing the IDE, including repository instances for ontologies, adherent to the MOF metamodel, with standardized query and manipulation language.

References

1. S.Hildreth, Rounding Up Business Rules. ComputerWorld Software. ID, (2005).
2. OMG , MDA Guide Version 1.0.1, (2003). http://www.omg.org/cgi-bin/doc?omg/03-06-01. Access in March/2010.
3. OMG"Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification. (2008) http://www.omg.org/spec/SBVR/1.0/PDF/. Access in March/2010.
4. G. Governatori, Z. Milosevic and S. Sadiq, "Compliance checking between business processes and business contracts," Proc. The 10th International Enterprise Distributed Object Computing Conference - EDOC (2006), Hong Kong, pp. 221–232.
5. S. Beller, "Deontic norms, deontic reasoning, and deontic conditionals," Thinking & Reasoning, Volume 14, Issue 4, November, pages 305 – 341 (2008).
6. ATLAS group, LINA, INRIA, "ATL: Atlas Transformation Language ATL User Manual - version 0.7, Nantes (2006).
7. OMG. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, version 1.0, (2008).
8. OMG, "MOF 2.0/XMI Mapping Specification, v2.1.1"., http://www.omg.org/cgi-bin/doc?formal/2007-12-01/. (2005) Access in March/2010.
9. Kamada, "Service Execution based on Business Rules", PhD Thesis in Computing Engineering, Unicamp, Campinas, Brazil, 2006.
10. Kamada, A. Figueiredo and M. Rodrigues, "Ontology based Business Rules and Services Integration Environment ," in Handbook of Research on E-services in the Public Sector: E-government Strategies and Advancements, Editors: Abid Thyab Al Ajeeli, Yousif A. Latif Al-bastaki, Jeihan Abu-tayeh, Information Science Publishing, March 2010.
11. G. Governatori and Z. Milosevic, "A Formal Analysis of a Business Contract Language," International Journal of Cooperative Information Systems (IJCIS), Volume 15, Number 4, pp.659-685 (2006).
12. G. Governatori. Representing business contracts in RuleML. Int. J. of Cooperative Inf. Sys., 14(2-3):181–216, 2005.
13. G. Governatori and A. Rotolo. Logic of violations: A Gentzen system for reasoning with contrary-to-duty obligations. Australasian Journal of Logic, 4:193–215, 2006.
14. S. Kanger. Law and logic. Theoria, 38:105–32, 1972.
15. V. Kabilan," Contract Workflow Model Patterns Using BPMN", FORUM 100, Kista, Sweden, 2005.
16. B.N. Grosof and T. C. Poon, "Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions," Proc. 12th International Conference on World Wide Web, Budapest, Hungary, 2003, pp. 340 – 349.
17. Abrahams, D. Eyers and J. Bacon. "An asynchronous rule-based approach for business process automation using obligations**,"** Proc. ACM SIGPLAN Workshop on Rule-based Programming, Pittsburgh, USA, 2002, pp. 93 – 103.