Flexible Software Development based on Business Rules

Aqueo Kamada[1,2], Manuel Mendes[2,3]

[1]CenPRA, Rod. Dom Pedro I, km 143.6, 13082-120, Campinas, SP, Brazil

[2]Unicamp, Cidade Universitária "Zeferino Vaz", 13083-970, Campinas, SP, Brazil

[3]Unisantos, Rua Dr. Carvalho de Mendonça, 144, 11070-906, Santos, SP, Brazil
aqueo.kamada@cenpra.gov.br, mj.mendes@terra.com.br

Resumo. *Atualmente, com as relações entre as pessoas e organizações baseadas em aplicações computacionais via Web, a necessidade de mudanças nestas aplicações acontece em períodos muito curtos. Neste contexto, este artigo propõe um modelo para capturar as mudanças nos negócios e rapidamente implementá-las nos sistemas computacionais. O modelo inclui uma abordagem para desenvolvimento rápido de serviços considerando que as porções voláteis da lógica de negócio possam ser externalizadas como regras de negócio e as porções estáveis como serviços Web. Apresenta-se um ambiente de desenvolvimento integrado de regras de negócio e ontologias como parte deste modelo, onde os fatos e as condições das regras de negócio são associados a serviços Web.*

Abstract. *In the current enterprise systems, with the relationships between people and organizations based on computational applications through the Web, the need of changes in these applications happens in periods shorter and shorter. In this context, this paper proposes a model that can capture business changes and quickly implement them into computational systems. The model includes an approach for fast development of services considering that the volatile portions of business logic can be externalized as business rules and the stable portions as Web Services. It is presented a business rules and ontology integrated development environment (IDE) as part of this model, where business rules' facts and conditions are linked to Web services.*

1. Introduction

Enterprise computational systems normally help to organize the data and to manage the daily activities of specific businesses. Hence, a computational system should reflect the knowledge of the business domain in which the system is used. Commonly, documents containing procedures, contracts, regulations and laws define the strategies, policies and relationships among organizations and consolidate that knowledge. From those documents arise the rules that define the behavior of the businesses in the organizations [Hildreth 2005]. Thus, a set of such rules, known as business rules, that contain important information for the realization of the business represent the business essence and define how an organization must behave to deal with its daily situations.

Due to the high competition, communication speed and degree of the customers' information, the organizations have to make changes in their computational systems in a much more accelerated rhythm than in last decades. In other words, the business rules have to be modified in short periods of time. Consequently, the computational solutions for the business problems cannot accompany the speed in which the change necessities appear. In this context, companies and governments need to incorporate new technologies to implement solutions integrating processes and services already existent with new processes to generate other services. In this way, all the time, business rules, processes and services need to be reviewed so that the organization can act according to the changes that happen in the business scenario. For most of the organizations the adaptation capacity to the new realities in a fast and efficient way is a crucial challenge to maintain their competitiveness or their survival in the market.

The difficulties and inflexibility to quickly reflect the business' changes into the computational systems motivated this research. The identified reasons for those difficulties are as follows: (i) in most of the computational systems the business rules are dispersed in the documentation and in the executable code and this turns the maintenance slow and expensive [Halle 2001] and [Hildreth 2005] and (ii) the orthogonality between the perspectives of the process composition mechanisms and the business rules mechanisms make the integration of these mechanisms difficult in terms of collaboration in treating the business logic [Charfi and Mezini 2004] and [Orriens and Yang 2006].

In the business and government applications there are portions of the business logic that are dynamic and sensitive to the market changes and other portions that are quite stable. However, the monolithic form in that most of the applications were implemented with the dynamic portions built-in in the computational code make difficult the changes implementation. These portions, dynamics and stable, need to be appropriately structured and maintained separate so that they can be updated independently. Then, these dynamic and stable portions need to be integrated to perform new services in the context of the fast changes that the market demands.

To answer quickly to these changes demanded by the market, this work proposes a development model with the following contributions: (i) a development platform composed of a set of tools to support the formal development of rules in the business people's terminology, and (ii) a set of repositories for vocabularies, rules and ontologies to guarantee the necessary support in terms of semantics, persistence and integrity for the manipulated artifacts.

The next section discusses some aspects related to business rules and Web Services and the convergence movement between them. Section 3 presents some related works and section 4 provides background overview of business rules. Section 5 provides a general idea of the proposed architecture emphasizing some aspects related to the development environment based on business rules. Section 6 presents an overview of an e-Government scenario to exercise the proposed ideas. The final section provides a conclusion and future researches.

2. Business Rules and Web Services Convergence

Recently, the researches and technologies related with business rules faced a new pulse due to their potential to capture and to facilitate the implementation of the changes that happen in the organizations' daily activities. Many business rules mechanisms [Fair Isaac 2007], [ILOG 2007], [Friedman-Hill 2003], the called business rules engines and Business Rules Management Systems (BRMS), which are well established, already demonstrate the effectiveness of their use in some sectors of the market, especially in the finance and insurance sectors. The business rules mechanisms guarantee flexibility by making possible that potentially all the decision points of process flows can be edited and stored in a managed rules repository. Thus, they make it possible that the applications can access the rules mechanism and the proper rules can be updated quickly by the business analysts. The most common problems in all of the rules mechanisms refer to the difficulty in the rules formalization in natural language associated with the respective execution support and to the difficulty in the combination of the business rules with other aspects of business processes.

On the other hand, the researches and the technologies related with the Service Oriented Architecture (SOA) [Barry 2003], [He 2003] also faced a strong pulse with the Web services popularization [W3C 2004] due to their potential in the integration and interoperation of distributed services, independently of location, development platform, execution environment and of programming language [Mahmoud 2005]. Also in this area the rules mechanisms can have an important role [Fair Isaac 2005]. The legacy applications present strong reasons for the organizations to experiment rules mechanisms and SOA technology. When the organizations have many built-in rules in a legacy application migrating them to rules mechanism and maintaining the stable processes as Web Services allow the users to make changes without having to rewrite code constantly [Cuecent 2006]. Again, the most common problems refer to the difficulty in the combination of business rules with Web services.

In the last few years the initiatives including business rules mechanisms in the context of the service composition machines, such as BPEL (Business Process Execution Language), have been contributing to facilitate the services composition process, although they do not have the ability to appropriately treat the business rules in a collaborative way with the execution of services [Nagl Rosenberg and Dustdar 2006], [Geminiuc 2006]. The Business Process Management (BPM) community is paying attention to business rules mechanisms, leading to partnerships between BPM suppliers and business rules mechanisms suppliers [Hildreth 2005].

3. Related works

Although it has not been identified in the literature any similar work, considering the cooperation between the development environment and the business rules execution environment in the task of discovering and executing services, there are some works related to the business rules execution. All the related works mention the absence of an appropriate treatment so that the Web services composition mechanisms work in a collaborative and integrated way with the business rules mechanisms. Hence, all of them confirm that, the technologies and existent standards for business computational systems development can not satisfy the demands to perform agile and dynamic businesses.

Charfi and Mezini propose the hybrid composition approach [Charfi and Mezini 2004], combining business rules extracted from Web services composition specifications described, for instance in BPEL. They discuss two technological alternatives to implement business rules in modularized units: one using Aspect Oriented Programming (AOP) concepts [Elrad, Aksit, Clarke and Filman 2004] and the other using rules machines. They defend that the approach allows a more flexible and modular way for Web services composition than the pure and simple use of Web services composition mechanisms based on processes, such as BPEL. The weakness of this alternative is that it presents a great open issue. This issue falls exactly in the most important aspect of the proposal: the integration of rules and processes. The authors alert for the fact that the simplicity of the user's point of view is accompanied by a high complexity in the underlying orchestration machine.

Bart Orriens and Jian Yang proposed a framework, called BCDF (Business Collaboration Development Framework) [Orriens and Yang 2006] that increases the flexibility and adaptability of computational systems, where business rules are used to drive and to restrict collaborations among business processes. They argue that the approach increases the flexibility due to the fact that the development of collaborations is driven by business rules, which can be chained and used to automate diagnoses and complex decision points. This proposal, although it still doesn't have a complete implementation and presents some open issues, such as the treatment of quality of service, it presents a sophisticated development model to increase the flexibility and adaptability of computational systems. An aspect that deserves improvements is the one that refers to the definition of the rules, which is not user friendly, because it must be described in RuleML [RuleML 2006]. Hence, it is very far from the terminology used by business analysts. It means that, in this proposal, the reduction or elimination of the semantic gap among the business people and software engineers' languages was not taken into account.

ILOG JRules 6 [Bloor Research 2006] is considered one of the most complete Business Rules Management System (BRMS) in the market. JRules 6 allows the use of the complete object model and XML Schemas. Thus, Java objects and XML are combined in the Business Object Model (BOM). Based on BOM models, the objects and their methods are translated into fragments of natural language. In this way, for instance, the Customer object can be associated to the fragment "customer" and the method *getAge* can be associated to the fragment "the age of". Considering this abstraction level, the following rule can be built using these elements in natural language: "**If** the age of customer **is greater than** *18* **AND**... **Then**... ", where, **text in bold** is used to show key elements of JRules 6 builder, underlined text is used to show the translation of BOM objects, and *text in italic* is used to show the user's input for JRules 6 builder. The advantages of this product are that it is already in a well stabilized version, it is based on the JSR 94 standard [Java community Process 2006], and it accepts rules described in natural language. The drawback is that, although the rules are described in natural language and the methods associated to the fragments can be operations of Web services, the associations need to be performed by software engineers. This happens because the business analysts don't have enough knowledge to perform the associations of terms and predicates with the respective objects and

methods. Therefore, also in that product, the reduction or elimination of the semantic gap among business people and software engineers' languages is not possible.

4. Foundations on Business Rules

This section presents the concept of business rule in the context of this work, some specific business rules examples related to specific communities and it discusses some aspects related to the OMG's SBVR metamodel [OMG 2006].

4.1 Definition of Business Rule

Although there are a lot of discussion around the definition of what "business rule" means [OMG 2006], [IBM 2004], 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.".

4.2 Business Rules Examples

Following are some examples of business rules that permeate the business organizations. It should be explained that the great majority of such business rules is not properly exposed and, therefore, is not under the control of business rules mechanisms. The lack of efficient mechanisms for business rules management is the root of the inflexibility problems when implementing the changes in the computational systems.

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.

Some business rules in the context of air ticket reservation:

- The price of an air ticket from São Paulo to João Pessoa is R$ 1000.00 if the flight departure and arrival happen in the middle of the week. And R$ 500.00 if the flight departure or flight arrival happen in the weekend.
- Every air ticket reservation will be lost if it is not paid until 48 hours before the flight time.

Some business rules in the context of civil identification:

- A citizen that lost the identity card needs to register a notification notifying the loss.
- An identity card should be canceled if a notification was registered by loss.
- The identity card should not be canceled if the data supplied in the notification are not same to the identity card.

4.3 SBVR Metamodel - Semantics of Business Vocabulary and Business Rules

Due to the lack of consensus to define rules in the terminology of businesses OMG (Object Management Group) published a Request For Proposal (BSBR - Business Semantics of Business Rules RFP) [OMG 2003]. This RFP requested proposals for the SBVR (Semantics of Business Vocabulary and Business Rules) metamodel [OMG 2006], which is in final phase of adoption by OMG. The main objective of the SBVR

metamodel 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. This RFP asked proposals for (i) a metamodel for the specification of business rules by business people, with a MOF (Meta Object Facilities) representation; (ii) a metamodel for the capture of vocabularies and definitions of the terms used in business rules; and (iii) an XML representation of business rules and vocabularies based on XMI to allow the interoperability of rules and vocabularies among software tools that manipulate business rules [OMG 2005]. The resulting SBVR metamodel is not intended for the business people, instead it is intended for software engineers that build tools for business people, because it is located in the CIM (Computation Independent Model) level of the MDA (Model Driven Architecture) [OMG 2003b], as suggested by OMG.

The foundation of business rules in SBVR has its origins in the idea presented in the "Business Rules Manifesto" [BRG 2003], where "Rules are based in facts, and facts are based on concepts that are expressed by terms. Terms express business concepts; facts make assertions on these concepts; rules restrict and support these facts." That basic idea, which appeared in 1995 in the context of Business Rules Group [BRG 2007], has been called of "mantra" of business rules. For convenience, that "mantra" is, frequently, abbreviated to "Rules are based in facts, and facts are based on terms." Figure 1, extracted of SBVR [OMG 2006] tries to show how the specification supports the idea of that "mantra."

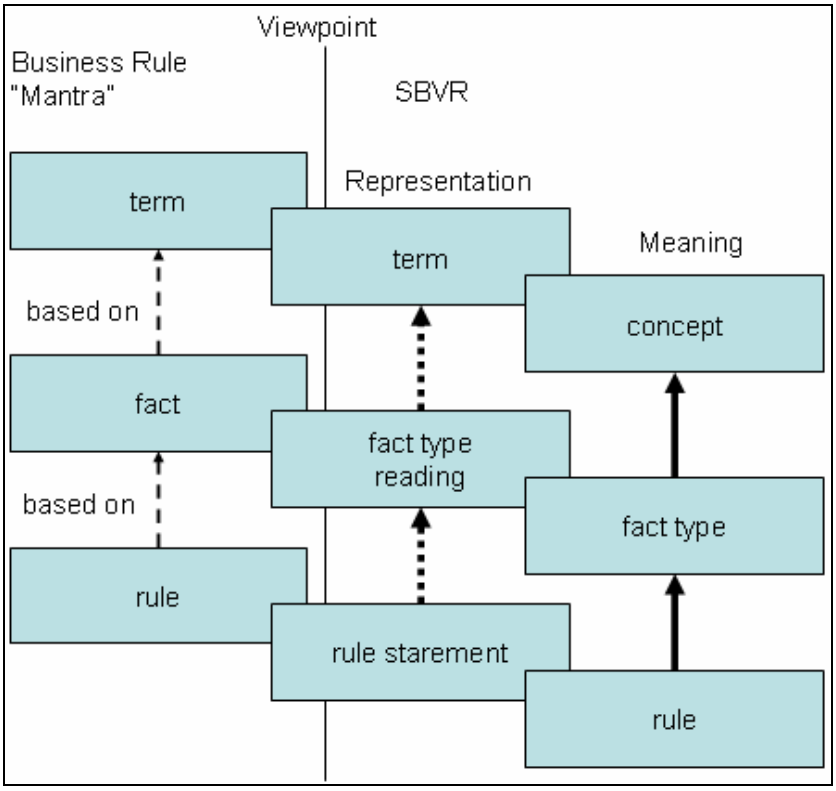

Figure 1. SBVR supporting the Business Rules "Mantra".

Business Vocabulary

An important aspect of business rules is that they can and they should be presented to business people in their own terminology. In other words, the business rules should be intelligible for business people that have the responsibility for the business activities for which the rules are applied, and for the people that understand how the rules link to the objectives of the business. The syntax of the rules can be textual or graphic, but the vocabulary should be a business vocabulary, significant to the people that do and perform the business rules, as specified in BSBR-RFP [OMG 2003].

5. The Services Model based on Business Rules

This section presents the proposed Service Development and Execution Model based on business rules that allow implementing quickly in the computational systems the changes demanded in the daily businesses of the companies and governments. The model is an evolution of the one proposed in [Kamada and Mendes 2005], [Kamada and Mendes 2005b], and [Kamada 2006]. Although an overview of the proposed model is provided, the main focus will be on the concepts for development of business rules.

Figure 2 displays the Service Development and Execution Model.

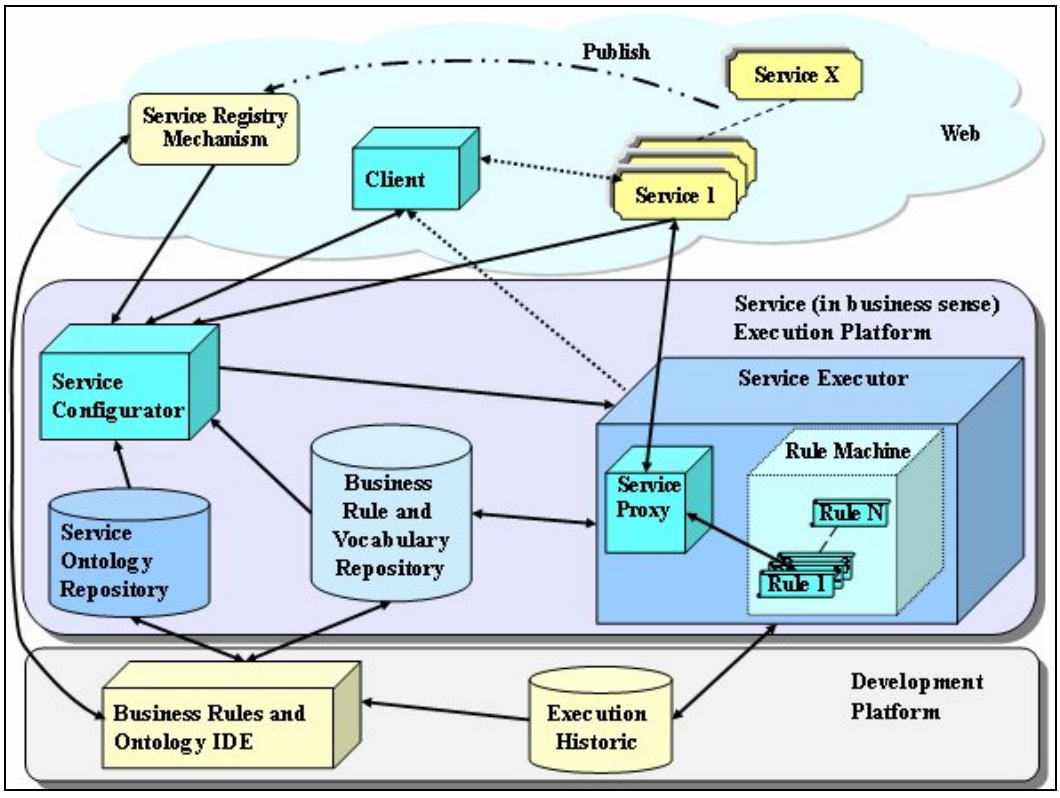

Figure 2. Service Development and Execution Model.

The Client module interacts with the Service Configurator to discover a Client need that, in practice, leads to the realization of a service. The Service Configurator obtains the elements related with the Client's description from the Business Rules Repository and Service Ontology Repository and, if it is the case, it interacts again to determine the Client need. These elements are consolidated in an entity nominated as

Service Profile that will be sent to the Service Executor. The Service Executor is responsible for recovering in the Business Rules Repository the pertinent business rules, instantiating a rules machine, translating the business rules, recovering the locators (also referred as, interaction point, endpoint or URL - Universal Resource Locator) of the Web services searching the Service Registry, generating proxies for the Web services, executing the business rules and returning the result to the Client. The elements of the repositories are created and managed by the IDE. The next subsection presents details of IDE, which is the main focus of this article.

5.1 Business Rule and Ontology Integrated Development Environment

The IDE is used to define and manage the elements (terms, verbs, qualifiers and questions that compose the vocabulary and the business rules) that are used by a business community. These elements should be precisely defined and structured in the repositories, so that the access is easy and consistent. The IDE accesses the Rules Repository and the Ontology Repository to update the business rules and the ontologies, at the same time that the Executor queries these repositories searching for rules to execute. Thus, an important aspect in the IDE is that it leans on the use of ontologies to structure the services, taking into account the business rules contained in the Rules Repository, the Web services published in the Registry and the business rules execution report.

The editor is the most important component in this IDE. The editor provides means to define and to maintain the elements that compose the vocabularies, the business rules and the ontologies. The proposed editor bases on the use of templates associated to the different types of existent rules. The SBVR metamodel defines some types of rules, such as those that treat obligations, prohibitions and needs. Taking advantage of Ross's ideas [Ross 2003] that propose the use of templates to facilitate the specification of business rules, this editor provides a group of templates associated to the types of rules defined in SBVR. In this way, for the modal type " is obligatory that" that treats of obligations the template has the following syntax: "It is obligatory [that] <fact> [if/while <condition>] ". Therefore, the IDE also includes an internal structure to store the templates and a set of operations for the template maintenance.

Although not always of immediate perception, some facts that define the type of the rule represent actions, whose realizations are associated to the execution of part of the service. Thus, some <fact>s that compose the modal type " is obligatory that" are translated in actions, that, once performed, they will have as result the assertion that the facts are true. This kind of action, in general, represents segments of the business logic quite stabilized, in the sense of little vulnerability to changes, so that the encapsulation of these actions as Web services becomes quite interesting. In this sense, the actions itself or the names of the rules give indications of the need or not of a Web service to perform such action. Therefore, in the creation or edition time of a business rule the Editor researches the Registry for such a Web service and makes a "early binding" between the business rule and the Web service associated to the actions. The early binding is quite providential because it avoids the risk of, in the execution time, not to meet a certain Web service that would perform part of the service. In this instant, the Editor joins information of this early binding connection in a structure nominated Ontology, associated to the service. This Ontology is stored in the Ontology Repository

so that, in the execution time the Executor, using the descriptions of the Web services, prepare the invocations.

The associations between the process Editor class and the several classes that represent the elements treated by it are displayed in Figure 3. The Editor contemplates the basic operations recommended by some design patterns to get, to set, to insert and to remove the elements under its control.

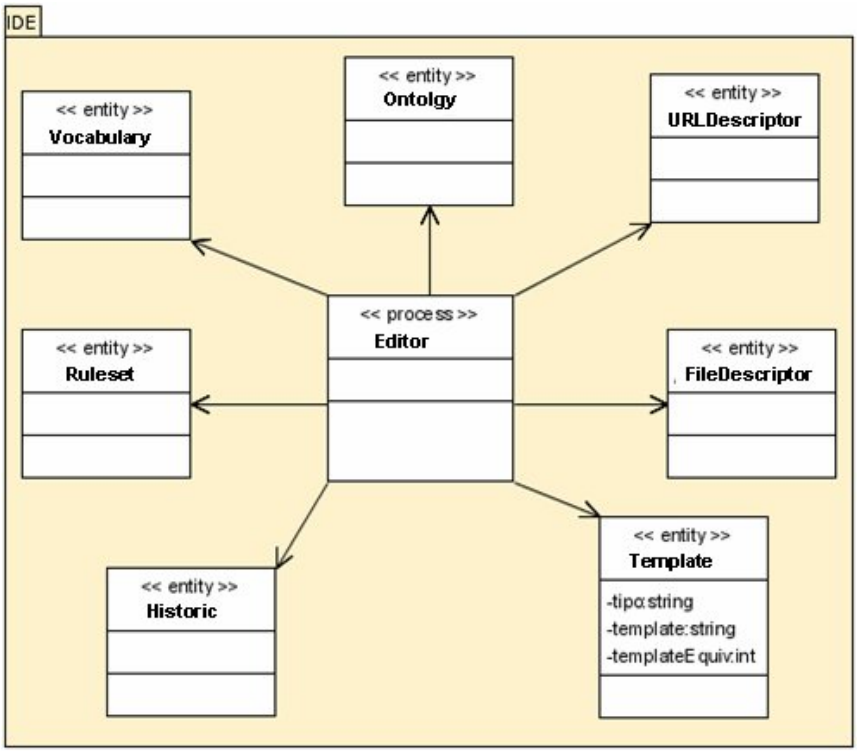

Figure 3. Editor Classes Diagram.

The definition of a rule involves several operations and bases on the use of templates associated to different types of business rules. So, to facilitate the development of business rules, those templates are incorporated in a graphic editor. Figure 4 shows an example of the interface prototype for this kind of editor.

Thus, to "Define Rule"

- The business analyst selects in the Editor the community's vocabulary (File→Open - identification.rul) and the set of rules from which the rule will be part (Rules Editor →Rule Set tab, selects the rule set and the eventual rules already defined will appear in the inferior panel).
- The business analyst selects the wanted template.
- The Editor gets the template and presents it in a window of the Editor.
- When the business analyst points to a field of the template the Editor gets the list associated to the field. For instance, in the template "It is obligatory [that] <fact> [if/while <condition>] ", pointing to <fact> or <condition> the list of predicates related to the selected vocabulary is obtained from the Rule Repository. Once the predicate is chosen and pointing to one of their fields the list of terms that make sense in that context is obtained in the Rules Repository.

- Some <facts> and <conditions> in order to become true presuppose that some actions have to be executed, which can be performed by a Web service. The business analyst selects "Search WS" and, with the words contained in a <fact> or <condition> the Editor executes "Search Descriptor URL" to discover if there is some Web service that performs the action.
- For each URL of valid WSDL file the Editor looks for the respective WSDL file.
- The Editor opens each WSDL file and obtains the element-keys, such as, URL of the Web service and the operation names. These elements are inserted in the Service Ontology to which the rule in definition should be associated.
- The business analyst can select "Check" to verify the consistence and the syntax.
- Finally, the business analyst selects "Insert" to insert the rule in the rule set that is being treated.

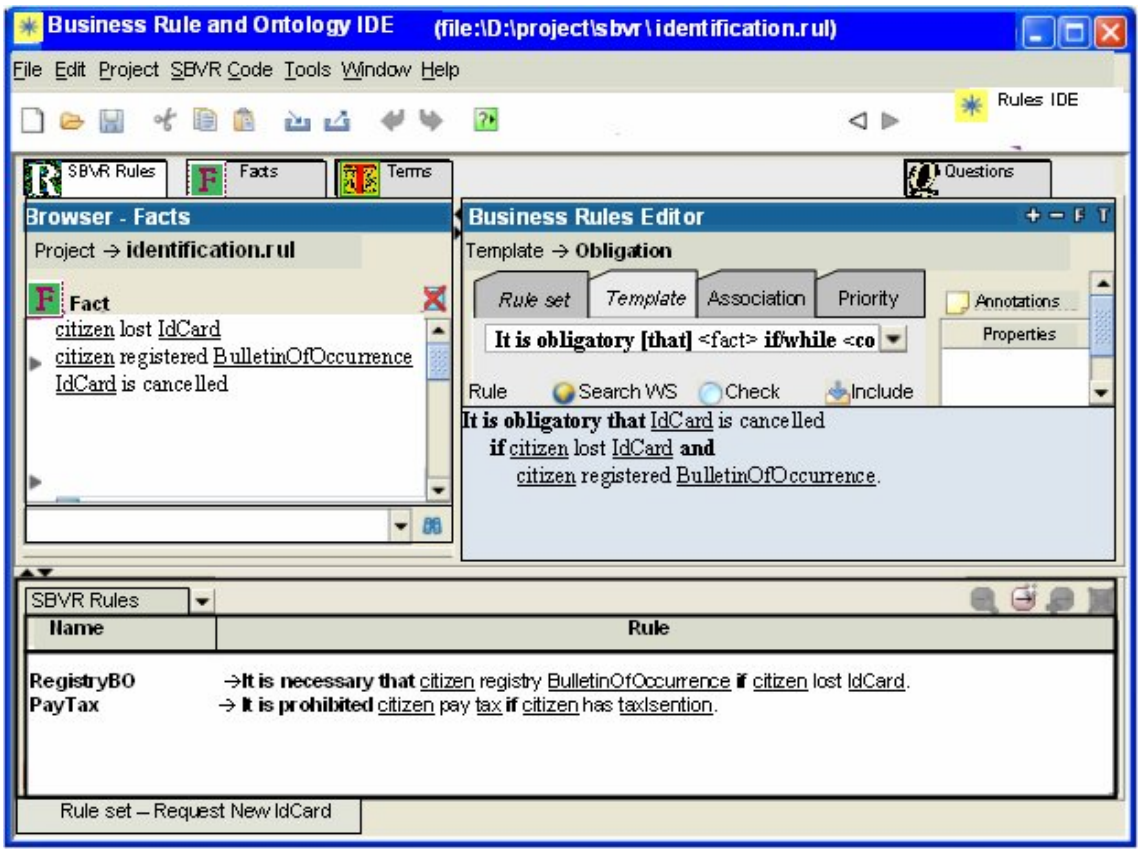

Figure 4. Interface Prototype for the Rule Editor.

5.2 Business Rules and Vocabularies Repository

The Business Rules and Vocabulary Repository instantiates a structure similar to Business Vocabulary+Rules, defined in SBVR [12]. Thus, it includes the business vocabulary and its respective set of business rules specified in terms of that business vocabulary. In other words, it organizes the terms, the concept definitions and the business rules that are specific to application domains or that make sense for a user community. It also organizes the relationships and the existent associations among these elements. To provide the ability in the definition of connections among concepts that are

of the organization interest the repository maintains a semantic structure in the business terminology.

Figure 5 displays the classes diagram for the Business Rules and Vocabulary Repository. A vocabulary is composed of four groups of elements, which are the terms, verbs, qualifiers and questions. The terms and the verbs are the fundamental elements of the vocabulary and the qualifiers and questions are elements included to facilitate the Service Configurator in the service discovery task.

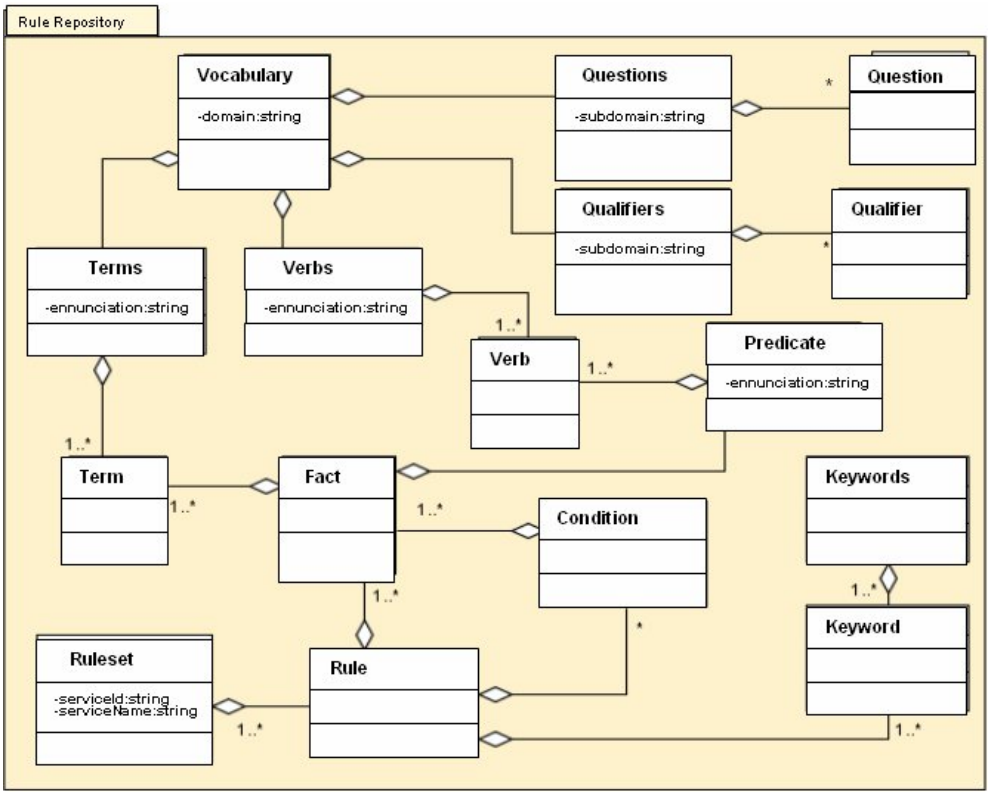

Figure 5. Class Diagram for Business Rule Repository.

A term represents a concept about an object and, according to SBVR metamodel, also represents an instance, which for extension is a fact, that it is also a thing. Thus, terms are used to represent anything, any concept, or any significant fact for a business community. The verb is the fundamental element of a predicate and, as such, it is the base for the representation of facts.

On the other hand, a Ruleset is a group of rules associated to a specific service. The rules are composed by keywords facts and conditions. Facts can be grouped in conditions. The keywords are very useful in the model creation to formalize rules without ambiguities. A fact is a simple period and, therefore, it is composed of one or more terms that can be associated (or not) to a predicate.

All these elements need to be properly organized in the repository so that the business rules are formalized referencing these elements consistently. Thus, rental and additional drivers, for example, are terms that designate a concept, which ultimately represent object types.

5.3 Service Ontology Repository

The Ontology Repository formalizes the relationships among names and terms and their respective meanings associated to the Web services. This formalization is explicit and specifies concepts, properties and relationships associated to the Web services, whose understandings are accepted and shared in the context of the community. In other words, the terms used to name the concepts, properties, and relationships capture the knowledge embedded in the specific community's jargon or the application domain. For instance, the services ontology for the finance community should formalize the names of the Web services as terms, and also the relationships among these names. Therefore, besides the terms as loan, interest rate, customer and credit history, there are also the names of the Web services that perform actions, such as, loan application, credit history verification and to customer registration that are fundamental to structure the Web services ontology for the finance domain. The class diagram for the Service Ontology Repository formalizes these relationships as can be seen in Figure 6.

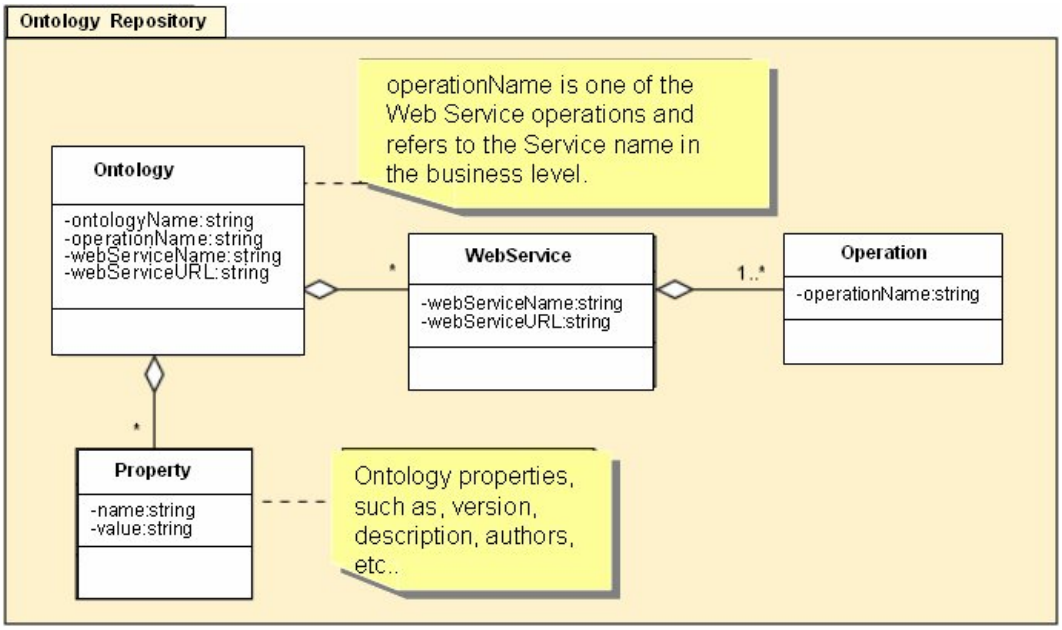

Figure 6. Class Diagram for Service Ontology Repository.

A Web services ontology structure takes into account that the realization of a Service demands the execution of a series of activities that are performed by different sections of an organization. These activities were automated in computational applications in different times and in different platforms and programming languages. With the current available technologies the encapsulation of these applications as Web services became an alternative with high demand in a lot of organizations [Software AG 2006]. That tendency reinforces the need of new mechanisms to integrate these Web services to create new services according to new requirements. Considering this, Web services ontology holds a set of properties of the ontology, a set of parameters of the Service and a set of Web services. Each Web service, for its turn, holds a set of operations with their respective parameters. Those relationships are fundamental elements in the Web services ontology because they represent the relationships among the terms that are, in this case, the names of the Web services, their operations and their

parameters. The instances of these relationship representations for a specific Web services ontology are generated by the IDE.

6. Services Usage Scenario

This section presents an overview of the e-Government scenario used in the proof of concept for the proposed service development and execution model. This scenario is inserted in the context of a Brazilian e-Government applications environment, in the São Paulo State [eGOIA 2006]. To simplify the scenario it is considered only some services related to the civil identification community and the specific life event of a citizen that had his/her identification card (IdCard) lost or stolen.

When a citizen has his/her IdCard stolen or lost he/she will go to a government Web portal in order to register the incident in an official form called BO (Bulletin of Occurrence) provided by a government agency through a back office system called BO System. Then, the IdCard Cancellation service provided by another back office system called Civil Identification System is accomplished to avoid the improper use of the IdCard. After that, the citizen can request for an IdCard replacement (a new issue), provided by these two legacy systems plus a set of five complimentary legacy systems. The issuing of new IdCard replacement issues may also be necessary any time and almost for any reason, such as, for example, when people are getting older or when people may not be recognized because of the old document's photograph, or even due to name change by marriage or even signature change.

Although the IdCard Replacement Request service is already offered to the population, it is not performed in an integrated and automated way as suggested in this work. The service was chosen because it presents a great variability of context data. This great variability of context data is due to the fact that the legacy systems that manipulate these data were built by different government's sections, in different times and using different platforms and development languages [eGOIA 2006]. For all these aspects, those legacy systems make available applications quite stabilized in terms of changes and, therefore, they are ideal to be wrapped as Web services.

6.1 Business Rules for Civil Identification Community

Adherent to the concept of rule set as a "collection of business rules grouped together for some purpose", included in the proposed SBVR metamodel [OMG 2006], are defined rule sets related to the Civil Identification Community.

The simplified rule set, contained in the Business Rules and Vocabulary Repository, related to the IdCard Replacement Request service could be formalized as shown in Table 1.

Table 1. Ruleset to IdCard Replacement Request service

Ruleset 1 - Rules to IdCard Replacement Request

- Rule 1.1 – **It is possible that the** citizen requests **an** IdCard replacement issue **if at least one of the following facts is true:**
  - **the** citizen lost **his/her** IdCard **and the** citizen registered **a** BO **and the** IdCard is cancelled
  - **the** citizen wants to change **his/her** photograph

- **the** citizen wants to change **his/her** name
- **the** citizen wants to change **his/her** signature

- Rule 1.2 – **It is obligatory that the** IdCard is cancelled **if all these facts are true:**
  - **the** citizen lost **the** IdCard **and**
  - **the** citizen registered **a** BO
- Rule 1.3 – **It is obligatory that the** citizen registers **a** BO **if one of the following facts is true:**
  - **the** citizen lost **his/her** IdCard
  - **the** citizen has **his/her** IdCard stolen

6.2 Executing Business Rules combined with Web Services

This section intends to show how actions should be dispatched when a business rule is in execution. In general these actions are implemented as Web services wrapping accesses to legacy applications.

Considering the Rule 1.1, once the fact (the citizen lost **his/her** IdCard) was already generated, the Service Executor discovers that it is necessary to execute actions in order to generate the facts "the citizen registered a BO" and "the IdCard is cancelled".

Based on the Rule 1.3, an action *to register a BO* should be dispatched, since the fact-condition "**the** citizen lost **his/her** IdCard" was satisfied. Assuming that the citizen accepted and supplied the necessary data to register a BO, the BO data begins to exist electronically and the fact "**the** citizen registered **a** BO" is generated.

As it can be seen in Rule 1.1, to request an IdCard it is necessary to have all of the conditions "**the** citizen registered **a** BO" and "**the** IdCard is cancelled" satisfied. At this point, the successful execution of the action to *cancel an IdCard* generates the fact "**the** IdCard is cancelled". Returning to Rule 1.1, as at least one of the pre-conditions is satisfied the action to *request IdCard Replacement* is dispatched to begin the process of requesting an IdCard replacement issue.

7. Conclusions and Future Researches

The premise of this work is that the use of the business rules technology combined with the emerging technologies derived from multiple researches areas provides an alternative solution much more flexible and promising than the traditional approaches for application development and execution. By externalizing the knowledge of the business that are built-in in the application procedures the business rules supply a solution possibility for the development of applications that need to be modified quickly. However, the existent rule mechanisms still cannot capture the rules in the business terminology and transform them in executable code without the software engineers' help.

The objective of this work was the proposition of a flexible model for development of services (in a wide sense), based on business rules, that can quickly implement in the computational systems the changes demanded in the daily activities of the organizations. The proposed business rules integrated development environment makes innovative contributions compared to other initiatives in business rules. The IDE

(i) helps business analysts in the definition of rules, using a language familiar to them, in other words, using the terms with which they accomplish their businesses; (ii) can capture the rules, using templates, and express them in computation independent models (CIM); (iii) helps business analysts in the definition of terms and typical predicates that compose the community's vocabulary, where the terms and predicates will be the base for the facts definition; (iv) renders the individualization and externalization of dynamic portions of the application logic as business rules; (v) makes possible to link the business rules with related Web services, in the instant of creation of the rule, excluding the need of the software engineer's intervention to do the connection.

The results indicate that the concepts, ideas and proposed platforms for the business rules development model are promising. Besides business rules technologies, services (SOA), repositories and ontologies, it seems that the complete solution for the mentioned problems includes the research to integrate other technologies, such as, model transformation, automatic program generation and compilation. Following are the 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.
- Consolidation of a method for automatic discovery of Web services based on the declaration of facts and conditions, during the execution of business rules.
- Inclusion in the Editor of another alternative to externalize business rules, for instance, using concepts of AOP, in similar way to the proposal in hybrid composition [Charfi and Mezini 2004].
- Definition and implementation of repositories, adherent to the MOF metamodel, with standardized query and manipulation language.

References

Barry, D. K. (2003) “Web Services and Service-Oriented Architectures: The Savvy Manager's Guide”, Morgan Kaufmann.

Bloor Research (2006) “ILOG Jrules 6.0”, White Paper, http://techfinder.itweek.co.uk/vnuitweek/search/viewabstract/10000003847/index.jsp

BRG (2003) “The Business Rules Manifesto”, http://www.businessrulesgroup.org/brmanifesto.htm.

BRG (2007) “Business Rules Group”, http://www.businessrulesgroup.org/home-brg.shtml

Charfi, A. and Mezini, M. (2004) “Hybrid Web Service Composition: Business Processes Meet Business Rules”, ICSOC'04, New York.

Cuecent. (2006) “Standard based Framework”, Cuecent BPMS. http://www.bahwancybertek.com/cuecentBpms.html Access in March/2007

eGOIA (2006) “Electronic Government Innovation and Access”, http://www.egoia.info/. Access in June/2007.

Elrad, T., Aksit, M., Clarke, S. and Filman, R. E. (2004) "Introduction to Aspect-Oriented Software Development", Addison Wesley Professional.

Fair Isaac (2007) **"**Enterprise Software for Business Rules Management", Fair Isaac Corp., http://www.fairisaac.com/fic/en/product-service/product-index/blaze-advisor/blaze-advisor.htm Access in March/2007.

Fair Isaac Corp. (2005) "Achieving Decision Consistency Across the SOA-based Enterprise: Using Business Rules Management Systems in an SOA", White Paper, http://researchlibrary.theserverside.net/detail/RES/1129644089_373.html.

Friedman-Hill, E. (2003) "Jess in Action", Manning Publication Co.

Geminiuc, K. (2006) "A Services-Oriented Approach to Business Rules Development", SOA Best Practices: The BPEL Cookbook, Oracle Technology Network.

Halle, B. V. (2001) "Business Rules Applied", Wiley, First Edition.

He. H. (2003) "What is Service-Oriented Architecture?", O'Reilly WebServices.XML.com.

Hildreth, S. (2005) "Rounding Up Business Rules". ComputerWorld Software. IDG.

IBM (2004) "Initial Submission to BSBR - Business Semantics of Business Rules".

ILOG (2007) "Deploying Rule Applications with ILOG Jrules", White Paper, http://www.ilog.com/products/jrules/whitepapers/index.cfm?filename=WP-JRules6Deployment.pdf Access in March/2007.

Java community Process (2006) "JSR-94 Java™ Rule Engine API", http://jcp.org/en/jsr/detail?id=94.

Kamada, A. (2006) "Service Execution based on Business Rules", PhD Thesis in Computing Engineering, Unicamp, Campinas.

Kamada, A. and Mendes, M. (2005) "Business Rule Engine Applied to eGovernment Services Integration", I3E / IFIP 2005, Poznan, Poland.

Kamada, A. and Mendes, M. (2005b) "Rule based flexible eGovernment applications", International Conference on E-Government, Ottawa, Canada.

Mahmoud, Q. H. (2005) "Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI)", SDN - Sun Developer Network.

Nagl, C., Rosenberg, F. and Dustdar, S. (2006) "ViDRE – A Distributed Service-Oriented Business Rule Engine based on RuleML", http://www.infosys.tuwien.ac.at/Staff/rosenberg/papers/TR/TUV-1841-2006-38.pdf

OMG (2003) "BSBR - Business Semantics of Business Rules RFP", br/2003-06-03.

OMG (2003b) "MDA Guide Version 1.0.1".

OMG (2005) "MOF 2.0/XMI Mapping Specification, v2.1".

OMG (2006) "SBVR - Semantics of Business Vocabulary and Business Rules", Adopted Specification. http://www.omg.org/docs/dtc/06-03-02.pdf

Orriens, B. and Yang, J. (2006) "Specification and Management of Policies in Service Oriented Business Collaboration", Tilburg University, Netherlands, Macquarie University, Sydney.

Ross R. G. (2003) "Sentence Patterns for Rule Statements", Principles of Business Rule Approach, Addison-Wesley.

RuleML (2006) "The Rule Markup Initiative", http://www.ruleml.org/.

Software AG (2006) "Unlock data and functions in your legacy systems", CrossVision.

Sotomayor B. (2005) "A short introduction to Web Services", University of Chicago. http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s02.html Access in March/2007.

W3C (2004) "Web Services Architecture", Working Group Note 11.