A Preliminary Approach to the Automatic Extraction of Business Rules from Unrestricted Text in the Banking Industry

José L. Martínez-Fernández[1,2], José C. González[1,3],
Julio Villena[1], and Paloma Martínez[2]

[1] DAEDALUS – Data, Decisions and Language S.A.
Avda. de la Albufera, 321
28031 Madrid, Spain
{jmartinez,jgonzalez,jvillena}@daedalus.es
[2] Computer Science Department, Universidad Carlos III de Madrid
Avda. de la Universidad, s.n.
29811 Leganés, Madrid, Spain
{joseluis.martinez,paloma.martinez}@uc3m.es
[3] Dept. Telemática, Universidad Politécnica de Madrid
ETSI Telecomuncación
28040 Madrid, Spain

Abstract. This paper addresses the problem of extracting formal statements, in the form of business rules, from free text descriptions of financial products or services. This automatic process is integrated in the banking software factory, permitting business analysts the formal specification, direct implementation and fast deployment of new products. This system is fully integrated with the typical software methodologies and architectures used in the banking industry for conventional development of backoffice or online applications.

Keywords: Business rules, banking industry, natural language processing, financial ontologies.

1 Introduction

Current trends in software development are paying special attention to Business Rules Systems (BRS), especially useful in environments where specifications are changing everyday as a reaction to market evolutions. A Business Rule can be defined as a statement constraining some aspect of the business, and a BRS is the system in charge of verifying the correct application of these Business Rules.

One of the market sectors where needs and requirements are suffering continuous changes is the financial and insurance industry. The ITECBAN (Architecture for Core Banking Information Systems) project is an initiative funded by the Spanish Ministry of Industry (INGENIO 2010 initiative, CENIT programme). The aim of this project is to develop a new core banking distributed platform.

E. Kapetanios, V. Sugumaran, M. Spiliopoulou (Eds.): NLDB 2008, LNCS 5039, pp. 299–310, 2008.

In order to make it easy the integration of Business Rules in nowadays Information Technology infrastructures, BRS not only include inference engines to verify the compliance with a Business Rule but also different tools to manage the creation, maintenance and sharing of these rules. One of the major weaknesses of these BRSs is the need of specific technical knowledge to be able to define and develop Business Rules. Usually, software developers are in charge of creating and maintaining Business Rules, attending requirements of business analysts. There has been a lot of interest in developing tools to remove this technical dependence, including the ability for business analysts to use natural language expressions to define Business Rules. As already mentioned, the development of these rules must be integrated in the overall software development cycle implemented in the target organization.

The process of extracting Business Rules from free text is part of a software tool called K-Site® Rules, whose aim is to support the development of Business Rules. The tool pursues three main goals: make any information system independent from the specific rule engine used to implement Business Rules; facilitate integration of Business Rules development process into the whole software development process of the organization or company; and allow business analysts to develop and deploy Business Rules without the support of software developers.

K-Site Rules is developed under a Model Driven Architecture (MDA) [7] to make it easier the integration with the other services and software components constituting the ITECBAN platform. MDA considers a computational independent model (CIM) where domain specifics must be described and business logic is covered. Under this level, there is a platform independent model (PIM) where a software system to support the CIM is described using one or more platform independent models (in our case described through the Unified Modeling Language, UML, the Semantic Web Rules Language, SWRL, to describe rules, Web Ontology Language, OWL, to define the domain and the JSR-94 standard [10] for the interaction with the rule engine). Finally, several platform specific models (PSM) must be specified to support the PIM level; at this level, concrete products, rule engine vendors in our case, are considered. Fig. 1 depicts the languages and tools selected to define the corresponding models according to the MDA architecture.

This paper focuses on the first level, the computationally independent model, more specifically, on the process to obtain expressions in the Semantics of Business Vocabulary and Rules (SBVR) language from unrestricted text. After this, SBVR sentences must be transformed into SWRL expressions. This later step will constitute the transformation between the CIM model and the PIM model used to describe business rules to be implemented by the system. The next section describes the method designed to identify Business Rules from text, including details about standards and technologies applied. The third section includes a description of the architecture designed for the system, identifying main elements and software components used. The following section is centered on the linguistic processing integrated as part of the system, included in the STILUS linguistic software platform. The fifth section covers the semantic modules introduced to reduce the aforementioned ambiguity and to relate each input text with a desired interpretation. Finally, some preliminary results are shown and some conclusions are depicted.

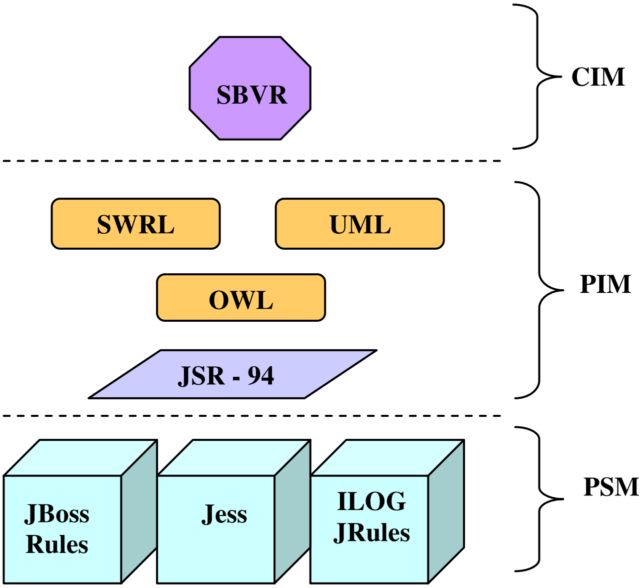

Fig. 1. Languages, standards and tools selected to build each level of the model driven architecture for K-Site Rules

2 Extracting Business Rules from Text

The process of extracting Business Rules from unrestricted text is supported by the SBVR standard and by the OWL language. The SBVR standard defined by the OMG group provides a way to document the semantics of business vocabulary, business facts and business rules. One part of this standard is devoted to the construction of several vocabularies, according to a hierarchical structure to be identified for the specific domain considered, to store concepts and relations among them. In the approach covered in this work, these vocabularies are going to be substituted by ontologies, either pre-existing (for example, the SUMO Finance ontology [8]) or specifically defined for the working domain. This representation will be later easily referenced by a SWRL expression of the Business Rule, which will also be expressed using OWL. On the other hand, tools to manage ontologies, like Protégé [6], can be used to maintain the vocabulary up to date. Although the notation provided by SBVR is not going to be directly applied, some clues provided by this standard will be applied to detect the presence of possible Business Rules in free text sentences.

2.1 Semantic Business Vocabulary and Rules, SBVR

This specification adopted by the OMG and defined by the Business Rules Group, BRG[1], tries to include semantics in the definition of the business and its governing

[1] http://www.businessrulesgroup.org

rules. For this purpose, SBVR provides two different vocabularies, one of them to describe business vocabulary, i.e., terms and their meanings apart from the ones appearing in the Business Rules, and the other one to make explicit the meaning related to a Business Rule, based on the business vocabulary. According to this specification, a Business Rule is “a rule under business jurisdiction”, i.e., it exposes the criteria needed to take some decision relating the business. Two kinds of rules are considered: structural rules, which express necessities of the business, and operative rules, expressing obligations to be fulfilled, or, as stated in [5], “rules that can be directly violated by people involved in the affair of the business”.

The definition of vocabularies in SBVR relies on three main elements: rules, fact types and concepts expressed by terms. A fact type can be seen as an association between two or more concepts, which are represented by terms. A rule is always constructed by taking a fact type and applying some obligation or necessity restriction. For example, the rule “a car must have at least four wheels” corresponds to an obligation restriction on the fact type “a car has wheels”.

The SBVR specification includes some annexes where the expression of SBVR structures in the English language is described. Here it is possible to find some keywords that are usually included in natural language Business Rules expressions, which can constitute some clues about where it is possible to find a Business Rule in a given text. For example, Table 1 shows keywords used to represent quantifiers.

Table 1. Example of keywords for quantifiers identified in the SBVR specification

Keyword	Quantifier
Each	universal quantification
Some	existential quantification
at least one	existential quantification
at least n	at-least-n quantification
at most one	at-most-one quantification
at most n	at-most-n quantification
exactly one	exactly-one quantification
exactly n	exactly-n quantification
at least n and at most m	numeric range quantification
more than one	at-least-n quantification with n=2

Another language that has been used as a source of linguistic clues to detect Business Rules is RuleSpeak [1] that makes some suggestions about the best way to express rules to assure a correct interpretation and understanding by business agents. The target language in the approach presented in this work is Spanish, so these keywords and cues have been transformed into their counterparts for the Spanish language.

2.2 The Semantic Web Rule Language (SWRL)

The initiative to include semantics in the World Wide Web initiated by Tim Berners-Lee, the so called Semantic Web, has produced a language to annotate web sites with meaning. This language is called OWL, which is related to the concept of ontology ("a specification of a shared conceptualization", [4]), where concepts, along with their valid interpretations, and relations among them are somehow represented. This language is proposed in our work to represent the business vocabulary that constitutes the basis to build Business Rules. Another initiative, sponsored by the W3C, has defined a language, grounded on OWL, to represent Business Rules, this language is called SWRL and it is based on RuleML, a language to express rules using Horn clauses. This language has two syntaxes, one of them XML based and the other one RDF based. According to [5], a rule has two parts, an antecedent (body) and a consequent (head), both of them constituted by a set of atoms that could be empty. An atom can be an OWL instance, an OWL data value, an OWL description or data range, an OWL property or an OWL built-in relation. In this way, rules can be referenced to existing concepts in an OWL ontology.

This language will be used to construct a platform independent representation for Business Rules. As described in the previous section, a transformation from CIM to PIM must be provided, i.e., a way to translate from a natural language expression of the rule to an SWRL representation. This is the purpose of the system described in this paper, to help in the identification of Business Rules present in a free text and to express them using the SWRL language. In this way, the automation of the rest of the process to implement the given Business Rule will be assured.

2.3 Business Rule Recognition Process

Taking into account the mentioned technologies, the process to get a free text and returning a set of possible Business Rules and their representation in SWRL is shown in the following example.

In the framework of the ITECBAN project, a use case was defined to test involved technologies. The use case is a functional document describing the way some saving products, offered by a financial entity, should work. Suppose that the starting point is a functional analysis document where it is possible to read the sentence:

> *"Si un Producto Ahorro es Deposito Financiero entonces la divisa asociada solo puede ser euro." [If a Savings Product is a Financial Deposit, then the associated currency has to be the euro.]*

Then, a process to detect if this sentence can describe a Business Rule is launched, a linguistic analysis is made, linguistic rules are applied to know if the sentence can contain a Business Rule and a typical if – then structure is detected. Besides, an ontology was specifically defined for this use case, and some of the concepts described in it were also detected in the sentence. As a result, the sentence is marked as a candidate to contain a Business Rule and the concepts residing in the ontology are also tagged. A final step is considered where a business analyst decides about the validity of the Business Rule and, if it is necessary, modifications to the rule sentence are made. Once the natural language content of the rule is available, a SWRL equivalent expression is automatically built to be able to continue with the Business Rule development process. Fig. 2 shows the final SWRL code for the rule given as an example.

```
<swrl:Imp rdf:ID="DivisaProductoAhorro">
    <swrl:head>
      <swrl:AtomList>
        <rdf:rest rdf:resource= "http://www.w3.org/1999/02/22-
          rdf-syntax-ns#nil"/>
        <rdf:first>
          <swrl:IndividualPropertyAtom>
            <swrl:argument1>
              <swrl:Variable rdf:ID="x"/>
            </swrl:argument1>
            <swrl:argument2 rdf:resource="#EURO"/>
            <swrl:propertyPredicate
              rdf:resource="#con_divisa"/>
          </swrl:IndividualPropertyAtom>
        </rdf:first>
      </swrl:AtomList>
    </swrl:head>
    <swrl:body>
      <swrl:AtomList>
        <rdf:first>
          <swrl:ClassAtom>
            <swrl:argument1 rdf:resource="#x"/>
            <swrl:classPredicate
              rdf:resource="#ProductoAhorro"/>
          </swrl:ClassAtom>
        </rdf:first>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:first>
              <swrl:ClassAtom>
                <swrl:classPredicate rdf:resource=
                  "#DepositoFinanciero"/>
                <swrl:argument1 rdf:resource="#x"/>
              </swrl:ClassAtom>
            </rdf:first>
            <rdf:rest rdf:resource=
              "http://www.w3.org/1999/02/22-rdf-syntax-
              ns#nil"/>
          </swrl:AtomList>
        </rdf:rest>
      </swrl:AtomList>
    </swrl:body>
  </swrl:Imp>
```

Fig. 2. Example rule expressed using SWRL

3 System Architecture

The system described in the previous section has been implemented according to the architecture depicted in Fig. 3. The linguistic analysis component is in charge of processing the input text to include morphological, syntactical and semantic information. The semantic data is based on compiled resources and on the ontology describing the application domain. The output of this analysis constitutes the input of the rule detection component, which is built on a set of linguistically motivated rules or patterns inspired in SBVR and RuleSpeak languages. The output of this rule detector component is supervised by a business analyst and then introduced in the rule transformation component, which provides an automatic translation from the natural language expression of the rule to its SWRL counterpart. It is worth mentioning that every concept referenced in the rule must be included in the ontology that supports the rule system.

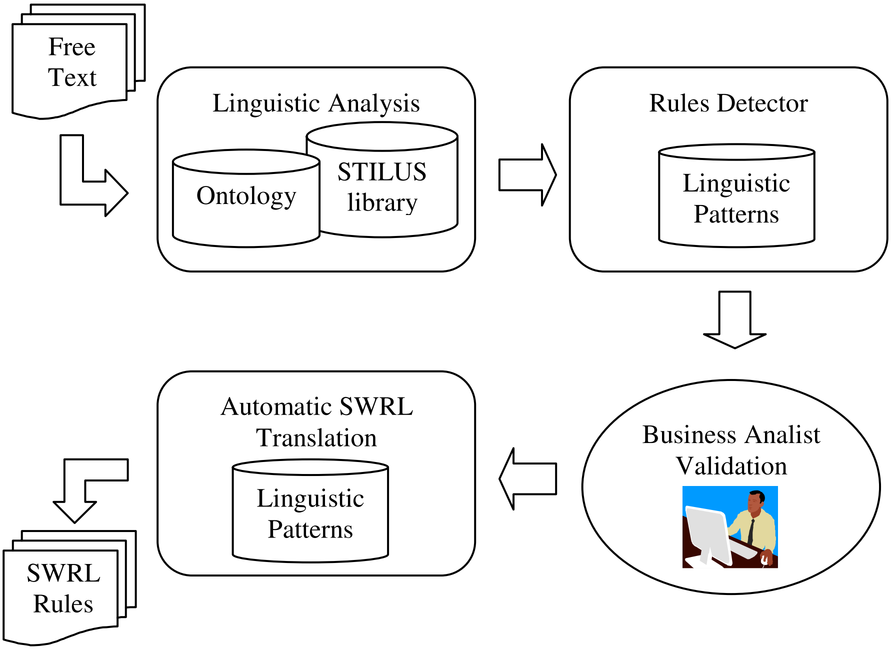

Fig. 3. Business Rules Extraction System Architecture

3.1 Linguistic Analysis

The linguistic analysis component is built based on STILUS®, a library for the morphologic, syntactic and semantic analysis of texts in Spanish. This software component, developed by DAEDALUS [2], is used to divide the input text in sentences, to provide the lemma for each word. To include information contained in the ontology, this lemma will be matched against the lexical realizations associated to each concept. On the other hand, each term in the document is related with the semantic tag present in the STILUS lexical base, if any. These data will be the basis for the rules detection component.

3.2 Rules Detector

This element of the architecture takes the linguistic information provided for the input text and applies a set of rules to decide if a sentence can contain a Business Rule or some clues to define one. This set of rules is based on some indications given by SBVR and RuleSpeak. A very simple example of these rules is *'if the sentence has an if – then structure then the probability for the sentence to contain a Business Rule is High'*.

3.3 Business Analyst Validation

The previous component has extracted a set of sentences containing a Business Rule, or part of one, with high likelihood. Of course, rarely these sentences are directly applicable as Business Rules, and human intervention is needed. This component includes necessary tools to show sentences to the Business Analyst and to retrieve valid natural language expressions for Business Rules according to SBVR specifications. The output of the component is then formed by valid SBVR expressions for Business Rules.

3.4 Automatic SWRL Translation

The final element considered in the architecture is devoted to the transformation between Business Rules in SBVR to Business Rules in a machine readable format, in particular, SWRL. The process followed to build the SBVR expressions assures that only concepts known by the organization, hence having some reflection in the available software infrastructure, are used to define rules. In fact, this software component is the one really devoted to the transformation from the CIM to the PIM in the MDA approach.

4 The STILUS Platform

STILUS is designed with a modular cascading architecture in which the output of each module may be attached to the input of the following one. This allows different versatile combinations to perform advanced text processing:

- **Text Segmentation:** segmentation is the process of dividing written text into words (usually called tokens) or other similar meaningful units, such as sentences or topics.
- **Part-of-speech tagger:** each word is assigned with its corresponding POS analysis. The morphological model is adopted from the ARIES platform [3]. The main idea is to automatically generate, for a given word, all possible root and derivative morphemes which can concatenate to each other according to the inflectional and derivational rules for Spanish. For example, root morphemes such as "hab-", "hub-", "hay-" are generated for the verb "haber", derivative morphemes "-o", "-a", "-os", "-as" for nominal gender and number inflection ("niñ-o-a-os-as"). In addition, the lexical database also includes other entries such as lexicalized words (those irregular forms that cannot be obtained with morphological inflection) and multiword expressions such as "a costa de", "Juan Carlos I", etc. Each unit has its own morphological information such as gender, number, person, verb tense, verb mode, word lemma, etc. The POS tagger uses this information to tag each word. The

lexical database is stored in a trie data structure that allows a very efficient read access.

- **Multiword recognition:** multiword units are identified by means of rules (such as dates or numbers) or linguistic resources (e.g. toponyms, film titles…)
- **POS disambiguation:** rules for morphosyntactic disambiguation that try to select the correct analysis for a given token among the proposed alternatives by the POS tagger. These rules cover linguistic patterns for specific words or combination of grammatical cases.
- **Syntactic parser:** this module turns a list of words into a syntactical tree with information about the type and function of each part of speech. This process consists of several steps including morphological and syntactical levels of analysis carried out in a bottom-up strategy. STILUS allows both a shallow parsing that simply identifies chunks (basic syntactic constituents of a phrase) and also a more complex analysis that computes chunks and the functional relations among them, thus tackling problems that have a semantic nature. The advantage of shallow parsing is in the case of ill-formed sentence, because the analyzer is still able to parse at least parts of the sentence.

4.1 Linguistic Rules

The linguistic knowledge is represented in a proprietary rule language specifically designed to abstract the linguists from the actual parser implementation and allowing them to focus on the linguistic phenomena.

The basic structure of a linguistic rule is:

```
IF
      conditions
THEN
      actions
END
```

"Conditions" include different test functions connected by boolean operators (AND, OR) and, if necessary, grouped into brackets. Table 2 shows different examples of test functions. For example, EXISTENTIAL_POS receives two arguments (token to evaluate and a regular expression for the POS tag) and is true if any of the word analysis fulfils the condition. In turn, UNIVERSAL_POS receives the same arguments but is true if all analysis fulfill the condition.

Table 2. Some examples of test functions

Function	Meaning
WORD(<pos>,<regexp>)	If the word matches the given regular expression
STARS_WITH_I (<pos>,<regexp>)	If the word starts with the given regular expression, case insensitive
EXISTENTIAL_POS (<pos>,<regexp>)	If any of the word analysis matches the given regular expression
UNIVERSAL_LEMMA (<pos>,<regexp>)	If all lemmas match the given regular expression

"Position" indicates the rule focus, i.e., the word(s) under inspection. Usually a generic position N and relative scrolling (… N-2, N-1, N+1, N+2, …) are used. In this case, the rule focus will move throughout the sentence, looking for a context that fulfils the given conditions. In addition, there are other position functions, for example, CHILD_POSITION, which allows testing syntactic structures (trees) instead of flat sentences, or FIRST_EXISTENTIAL_POS_POSITION, which uses a predicate to look for the position of the first token that fulfils the given condition.

Last, "actions" comprise a set of operations that may be applied over the sentence. Some examples are shown in Table 3.

Table 3. Some examples of action functions

Function	Meaning
JOIN_SYNTAGM (<pos1>,<pos2>,<tag>)	Create a new part of speech joining the words from <pos1> to <pos2> and assigning POS tag <tag>.
SELECT_TAG (<pos>,<regexp>)	Disambiguate the given word filtering out the tags that do not fulfill the given regular expression
ERROR(<pos1>,<pos2>, <type>,<msg>)	Marks the context from <pos1> to <pos2> with an error

5 Preliminary Experiments

The system developed and described in Section 3 has been executed on the use case defined in the framework of the ITECBAN project. As already mentioned, this use case describes the way that some saving products must be managed when customers of a banking entity want to work with them. For this use case, the ITECBAN Business Rules team has created an ontology containing concepts defined by the bank. Furthermore, an input document with the functional analysis for the application, written in natural language, has been studied and Business Rules have been identified and written in an SBVR version for Spanish. The document is formed by a total of 216 sentences, from which forty eight Business Rules have been extracted. The input to the system is formed by a file with the functional analysis document, written in Spanish, and another file with the ontology (in OWL format) that has been defined for the use case. The output is composed by an annotation, in XML, of the input text, including a score that can be interpreted as the likelihood for the sentence to represent a Business Rule. This score ranges from 0, when the sentence does not include a Business Rule, to 10, when it is almost sure that the sentence can be used to define a Business Rule. At this point, there is a validation step with human intervention to confirm if there is a rule in the sentence and, probably, to re-write it in a clearer expression using the SBVR language. Fig. 4 shows two examples of the system output (before the human validation step).

Table 4 shows some results produced, comparing the total number of sentences, the number of sentences marked by the system as containing a Business Rule (or part of it) and the number of Business Rules produced by hand by the Business Rules team of the project.

```
  <frase puntuacion='5'> Cualquiera de los padres/tutores puede
efectuar  cualquier  movimiento  que  admita  el  producto  sin
necesidad   de   contar   con   la    firma   del   resto   de
padres/tutores.</frase>

  <frase puntuacion='8'> La titularidad y divisa de la  cuenta
asociada  deberán  ser  idénticas  a  las  del  contrato  que  se
apertura, y no podrá cancelarse mientras no haya finalizado la
operación inicial y sucesivas, en su caso.</frase>
```

Fig. 4. Examples of the output of the Business Rules Detector system

Table 4. Preliminary results for the Business Rules Detector system

Total number of sentences	Number of handmade Business Rules	Number of sentences possibly containing a Business Rule
216	48	42

It can be surprising noticing that the number of sentences automatically detected to contain a Business Rule is smaller than the number of handmade Business Rules. This is normal because there is no a one to one relation between a sentence and a Business Rule, i.e., a sentence can be broken down into several Business Rules, and, of course, repeated Business Rules must be deleted.

Of course, this is only a preliminary evaluation and there must be and will be a further evaluation carried out with the business analysts and developers that will act as final users.

6 Conclusions

Natural language specification of products, services, procedures, regulations, etc. in business environments tends to use simple and unambiguous standard language conventions. In particular, typical sentences make use of an implicit world vision (an ontology) and express definitions, concepts and restrictions with a clear associated semantics. On the other side, there is a pressing need for companies to develop new products and services at the fast pace imposed by global markets. The only way to achieve this is by giving business experts a central role in the whole software-based product development life cycle. The tools developed in the framework of the ITECBAN project are intended to assist business analysts in the specification, implementation and deployment of business applications, with minimum support from IT specialists. Although this work is still in an early stage, and full experimentation and evaluation has to be completed, evidence has collected about the interest and viability of this approach in a real environment.

Acknowledgements

This work has been partially supported by the Spanish Center for Industry Technological Develpoment (CDTI, Ministry of Industry, Tourism and Trade), through the project

ITECBAN (Architecture for Core Banking Information Systems), INGENIO 2010 Programme, and the project "Software Process Management Platform: modeling, reuse and measurement" (TIN2004-07083). Other partners in ITECBAN are INDRA Sistemas, CajaMadrid, Sun Microsystems and Grid Systems.

References

1. Business Rule Solutions: BRS RuleSpeak® Practitioner's Kit. Business Rule Solutions, LLC. PDF (2001-2004), http://BRSolutions.com/p_rulespeak.php
2. DAEDALUS, Data, Decisions and Language, S.A, http://www.daedalus.es
3. González, J.C., Goñi, J.M., Nieto, A.F.: ARIES: a ready for use platform for engineering Spanish-processing tools. In: Digest of the Second Language Engineering Convention, London, October 1995, pp. 219–226 (1995)
4. Gruber, T.R.: A translation approach to portable ontology specification. Knowledge Acquisition 5(2), 199–220 (1993)
5. Horrocks, P.F., Patel-Schneider, H., Boley, S., Tabet, B., Grosof, M.: Dean: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, draft 0.5 (November 2003), http://www.daml.org/2003/11/swrl/
6. Knublauch, H.: Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protégé/OWL. In: International Workshop on the Model-Driven Semantic Web, Monterey, CA (2004)
7. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1, OMG (June (2003), http://www.omg.org/docs/omg/03-06-01.pdf
8. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Welty, C., Smith, B. (eds.) Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine, October 17-19 (2001)
9. OMG: Semantics of Business Vocabulary and Business Rules (SBVR), First Interim Specification (March (2006), http://www.omg.org
10. Selman, D.: Java Rule Engine API Specification JSR-94. Draft 1.0 (2002), http://jcp.org/en/jsr/detail?id=94