TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
Baltic Journal on Sustainability
2010
16(3): 471–486

ONTOLOGY AXIOMS FOR THE IMPLEMENTATION OF BUSINESS RULES

Diana Kalibatiene[1], Olegas Vasilecas[2]

Vilnius Gediminas Technical University,
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania
E-mails: [1]diana@isl.vgtu.lt; [2]olegas@isl.vgtu.lt

Received 8 January 2010; accepted 5 August 2010

Abstract. Modern information systems (IS) have widely penetrated into many kinds of social activities by helping to perform business operations and to interact with the surrounding environment. Since business is under jurisdiction of government's laws, business mission and polices, guidelines and other documents, these business restrictions should be implemented in a business supporting IS and effectively managed. A business rule approach is widely accepted in IS community as a way to express different types of business restrictions and constrain operation of components of an IS. Nowadays, there are number of methods and approaches to extract, define and implement business rules. However, there is no a commonly accepted standard for their modelling. In this paper an ontology-based method for business rules modelling and implementation in the software of IS is presented. It allows to automate the transformation of rules from ontology to the software level and to reduce the cost of rules implementation and likelihood of errors. A case study is described in the paper also.

Keywords: business rule, ontology, axiom, transformation.

Reference to this paper should be made as follows: Kalibatiene, D.; Vasilecas, O. 2010. Ontology axioms for the implementation of business rules, *Technological and Economic Development of Economy* 16(3): 471–486.

1. Introduction

Modern information systems (IS) have widely penetrated into many kinds of social activities by helping to perform business operations. A number of authors analyses possibilities of applying IS in new areas, like monitoring of water resources (Dzemydienė *et al.* 2008), construction industry (Kaplinski 2009), civil engineering, etc. Since business is under jurisdiction of government's laws, business mission and polices, guidelines and other documents, these business restrictions should be implemented in a business supporting IS and effectively managed. A business rule approach is widely accepted in IS community as a way to express

ISSN 2029-4913 print/ISSN 2029-4921 online
doi: 10.3846/tede.2010.29
http://www.tede.vgtu.lt

different types of business restrictions and constrain components of an IS. Nowadays, there are number of languages, like Unified Modelling Language (UML) (OMG 2005) with Object Constraint Language (OCL) (OMG 2003), structured English (OMG 2008), etc., methods and approaches, like Demuth Hussmann, Loecher method (Demuth *et al.* 2001), the Ross method (Ross 1997), CDM RuleFrame (Boyd 2001), etc., to extract, define and implement business rules. Selected IS development methodologies are analysed and compared according to the business rule modelling possibilities in (Herbst *et al.* 1994). Authors concluded that the existing methodologies and languages are not suitable for modelling all types of rules.

Since ontology is suitable to represent business domain knowledge, we consider that ontology should be used in the process of business rules modelling. The reasons of this hypothesis are the following:

- Ontology presents not only business concepts and properties organised in a structure, but it is also presents consolidation axioms (Guizzardi *et al.* 2002) that constrain a structure.
- Derivation axioms are used to derive new information from the information encoded in the ontology.
- Finally, as stated in (Guarino 1998), ontology axioms (and ontology as a whole) are typically expressed in a formal way. For this reason, they can in principle be transformed to business rules automatically.

In this paper an ontology-based method for business rules modelling and implementing in IS is presented.

The paper is structured as follows. Related works are analysed in Section 2. An ontology-based modelling of business rules is proposed and described in Section 3. Section 4 describes a case study of applying the proposed method for the modelling of qualification requirements for teachers. The developed Axiom2SQL prototype is presented also. Finally, Section 5 concludes the paper.

2. Theoretical background and related work on business rule and ontology

In order to address the main problems of business rules modelling, this section presents a short investigation of the concept of a business rule, their classification according the implementation perspective and main problems of business rules modelling. The concept of ontology is also analysed here.

According to (OMG 2008), business rule is rule, which is under business jurisdiction. However, it is reasonable mention to mention levels of a business rule usage. From the computerisation perspective, three levels of abstraction can be defined: business system, information system (IS) and software system. A business system represents the business, where business rules arise. IS supports a business system by providing necessary information for business. Business rules are implemented in an IS as information processing rules. A software system supports an IS by providing necessary software for IS. Information processing rules are implemented in a software system as executable rules. Therefore, for the sake of simplicity the term “rule” is used in the paper.

At the business system, rules are statements that define or constrain particular aspects of a business in a declarative manner. For example, a customer could not buy more than his / her credit limit permits. At the IS, rules are statements that define information processing rules using a rule-based language, like OCL, etc. Expressions of information processing rules are very precise; i. e. terms used in expressions are taken from the particular data model (Hay 2003). For example, the following OCL expression "`context c: Company inv enoughEmployees: c.numberOfEmployees > 50`" constrains that the number of employees in the Company must always exceed 50. At the software system, rules are statements that define executable rules using a language of a specific execution environment, such as Oracle 10g[1], Microsoft SQL Server 2008[2], ILOG JRules[3], etc.

From the implementation perspective, all rules can be classified to (Valatkaite and Vasilecas 2004; von Halle 2002):

- **Structural rules** (terms, definitions, facts, and integrity constraints), which introduce definitions of business entities and describe connections between them. They can be implemented by a conceptual data model of a business, e. g. entity-relationship (ER) and UML class model. Therefore, terms, definitions, facts can be regarded as concepts not rules. Integrity constraints can be implemented by conceptual data model integrity constraints, like referential integrity constraints, cardinality constraints, and mandatory constraints and expressed, like OCL invariants. At software system level, integrity constraints can be implemented like SQL assertions, checks, and foreign keys.
- **Dynamic rules**, which can be expressed by event-condition-action (ECA) rules and implemented, like SQL triggers and SQL views (for the case of some derivation rules). A dynamic rule is: a *dynamic constraint*, which restricts transitions from one state of the business to another; a *derivation rule*, which creates new information from existing information by mathematical calculating or logical conclusion from facts; and a *reaction rule*, which tests a condition and upon finding it true performs a defined action.

Since implementation of structural rules is defined quite precisely (it can be seen from the definitions of integrity constraints in a conceptual data model, like CHECK, DOMAIN, NOT NULL, referential integrity and other constraints), we concentrate on the implementation of dynamic rules in the paper.

Ceponiene *et al.* (2009) in their paper addresses the problem of modelling – separating business process rules from business constraints in UML&OCL models as coupling them together require changing business processes when rules are changing, and vice versa.

The results of survey presented in (Zacharias 2008) shows that a) a large part of rule-based systems are created without any specific development process, b) almost half of the respondents use an integrated development environment (IDE) (such as the Ontostudio[4], Ilog Rule Studio[5], the Visual Prolog IDE[6] or the Semantic Web Rule Language (SWRL) tab

1 http://www.oracle.com/technology/software/products/database/index.html
2 http://www.microsoft.com/sqlserver/2008/en/us/overview.aspx
3 http://blogs.ilog.com/brmsdocs/2008/06/15/ilog-jrules-6-for-architects-and-developers-2/
4 http://www.ontoprise.de/en/home/
5 http://www.ilog.com/products/businessrules/
6 http://www.visual-prolog.com/

of Protégé (Hassanpour *et al.* 2009) that allows to edit, load, debug and run rules. For editing rules the most widely used tools are still textual editors (33%), a simple text editor or a textual rule editor with syntax highlighting (28%) and graphical rule editors (26%); c) verification is dominated by testing (90%) and code review (78%). 74% of respondents do testing with actual data, 50% test with contrived data. Advanced methods of test organisation are used by a minority, with only 31% doing regression testing and 19% doing structural testing with test coverage metrics.

None of the proposed languages or methods for business rules modelling has been accepted as technology standard yet. Only few of them deal with reuse of knowledge acquired in the analysis of a particular business domain and automatic implementation of rules. For example, Gudas (2009) proposes the Knowledge-based Enterprise framework for the analysis of knowledge management and development of an Enterprise knowledge base. However, not enough attention is paid for business rules.

The process of modelling business rules involves two main problems – determining the business rules (their acquisition from the business) and developing ECA rules (their implementation).

First of all, it is necessary to determine the rules of a business and ensure that they are appropriate. The process of acquisition which business rules apply to a particular situation often involves an open-ended search through multiple sources (Ross 2003). The consensus from all the business stakeholders should be obtained on the problem of which business rules should be used. When the business changes the business rules should be properly adapted to new conditions. Capturing, documenting and retaining of the business rules prevent the loss of knowledge, when employees leave an enterprise.

After the business rules are identified, it is necessary to develop ECA rules to implement them by information processing rules and corresponding executable rules. However, in a business domain or an ontology, to which the rules belong, they are not always expressed in terms of ECA rules. A part of these rules have explicit or implicit condition and action parts. The missing condition can always be substituted with a default condition state as TRUE. A part of rules may have no explicit action, since they can state what kind of transition from one system state to another is forbidden. But the majority of these rules do not define the event explicitly or implicitly.

The management of business rules using swarm intelligence is analysed in (Nenortaite and Butleris 2009). Authors propose the decision-making model, which is based on the application of Artificial Neural Networks and Particle Swarm Optimization algorithm. Kapocius and Butleris (2006) propose the repository model, which supports the storage and management of all components of the captured requirements, including functions, business decisions, data sources, conceptual data model elements, business rules and their templates. According to this observation, the following business rule modelling problems may be defined:

- It is difficult to identify business rules in a business domain, since they are informal and merged with other business text.
- It is difficult to classify business rules, since there is no commonly approved classifier.
- An identified set of business rules is inconsistent, incorrect and incomplete or redundant.

– It is not clear how to implement business rules, because of their variety.

– It is not clear how to manage the set of implemented rules.

The big amount of works on business rules acquisition from a business domain and implementation in IS shows that this is an important and relevant topic in IS development. However, the lack of a standard method or a language for business rules modelling in IS development means that it is not a straightforward problem.

Methods and languages proposed to model and implement business rules can be classified according to their drawbacks as follows:

1. *Non-existence of any graphical notation* – languages and methods of this category do not have any graphical notation. OCL, proposed to express dynamic rules for UML diagrams, and all OCL-based languages and methods, like the method presented in (Badawy and Richta 2002), CDM RuleFrame environment (Boyd 2001) used by Oracle[4], (Armonas and Nemuraite 2009) have no graphical notation.
2. *Non-explicit implementation* – languages and methods of this category do not deal with a way rules be implemented (automated, semi-automated or manual). It is expected that many rules specified by the proposed language will likely be enforced in an automated way; and in such cases, the semi-formal or formal language or method is proposed. The Ross method (Ross 1997), which proposes specific constructs for each of the rules families, rule templates presented by (von Halle 2002) and OMG proposed "*Semantics of business vocabulary and business rules*" (SBVR) (OMG 2008), which proposes to use logical formulations of rules with abstract, language-independent syntax for capturing the semantics of a body of shared meaning, can be referred to this category.
3. *Limited type of rules* – languages and methods of this category are limited on modelling a specific type of rules. Templates, presented by (Demuth *et al.* 2001), are developed to generate SQL views and triggers, but the trigger action part is not automatically generated. A method presented in (Badawy and Richta 2002) is suitable to generate triggers from consistency rules defined using OCL, but the authors limit the usage of method to consistency rules only.
4. *Suitable for rules implementation at the lower levels of abstraction* – languages and methods of this category deals with implementation of rules expressed in a formal way. These methods do not deal with elicitation of rules from the application domain. Their use rules already expressed in a particular formal language. Authors of (Armonas and Nemuraite 2009) define the transformation of OCL expressions to corresponding SQL code.

Ontologies in nowadays are widely used in the process of development of IS, since they are suitable to represent business domain knowledge. The semantic content expressed by the ontology (or ontologies) gets transformed and / or translated to an IS component assuring the *ontological adequacy* of the IS. Moreover, ontologies expressed in a formal way can be automatically transformed to IS components, like conceptual data model or rules, by reducing costs of manual development of IS components and likelihood of errors.

According to the (Vasilecas *et al.* 2009) paper, ontology defines the basic concepts, their definitions and their relationships, comprising the vocabulary of a business domain, and the axioms for constraining relationships and interpretation of concepts. Some authors, like (Falbo *et al.* 1998), distinguish properties from concepts also. According to the content of a business domain knowledge, ontologies can be: *lightweight*, which describes a hierarchy of concepts related by particular relationships (e.g., is-a, part-of, etc.); *light heavyweight*, in which constraints are added to restrict values of concepts and relationships, like cardinality constraints, possible length, etc.; and *heavyweight*, in which suitable axioms are added in order to express and restrict complex relationships between concepts and to constrain their intended interpretation.

The most existing ontologies, like WordNet (WordNet 2006), which can be used as a lexical ontology, Protégé ontologies (not all), ontologies presented by (Culmone *et al.* 2002) and (Lin *et al*. 2001), DBpedia (Bizer 2008), are lightweight or light heavyweight, since those have no axioms. In heavyweight ontologies axioms defined in a framework of a description logics (McGuinness and Patel-Schneider 1998), in some kind of logic language, like KIF (Genesereth 2006) in Protégé[7] ontology (Noy *et al.* 2000) and SUMO[8], or SWRL, which is an expressive OWL-based rule language, in Protégé ontology.

According to the purpose, we classify axioms in epistemological, consolidation, and derivation axioms (Vasilecas *et al.* 2009). *Epistemological axioms* are defined to show constraints imposed by the way concepts are structured (like, is-a relation, part-of relations, cardinality constraints, etc.). *Consolidation axioms* exclude unintended interpretations over the structure of the ontology specification. Finally, *derivation axioms* allow new knowledge to be derived from the previously existing knowledge represented in the ontology. Mostly, epistemological axioms are implemented by structuring concepts in an ontology. Consolidation and derivation axioms are not distinguished and are implemented using some suitable languages, like Protégé Axiom Language (PAL) (Grosso and Crubezy 2008), Ontology Web Language (OWL) (OMG 2005a) or SWRL.

According to the production rules, axioms are classified to *state* or *condition-state* axioms. State axioms define admissible states of a business domain. Condition-state axioms define admissible states of a business domain under condition. Axioms hold in a domain in all cases. However, computer systems should have information when they apply rules. Therefore, it is necessary to define important events and link them with corresponding rules during transformation of ontology axioms to information processing rules or executable rules.

3. An ontology-based method for business rules modelling

In this section we present the ontology-based method for business rules modelling.
The main steps of the applying the method are as follows:

1. Check if axioms are in an ontology. It warranties that axioms are in an ontology. Otherwise, a user should define axioms. *Note* that the creation of an ontology is not analysed here, since, it is not the topic of the paper. The method is based on the assumption that a user of the method has a necessary ontology.
2. Find an axiom.

7 http://protege.stanford.edu
8 http://www.ontologyportal.org/

It is searching for an axiom in an ontology.

3. Transform an axiom to a corresponding ECA rule:

If an axiom is found, its transformation to an ECA rule starts. The transformation of an axiom to ECA rule consists of 3 main steps: defining an event, defining a condition and defining an action. First of all, an action is defined.

3.1. define an event of an ECA rule as *insert*, *update* or *delete*;

At the second step, the type of an axiom is determined. It determines the type of an ECA rule: dynamic constraint in the case of a consolidation axiom or derivation constraint in the case of a derivation axiom.

3.2. determine the type of an axiom – is it consolidation or derivation axiom?

3.2.1. in the case of a consolidation axiom:

Note that a consolidation axiom can be a state axiom or a condition-state axiom. However, in the bought cases it is transformed to the condition of an ECA rule.

3.2.1.1. transform an axiom to the corresponding condition of an ECA rule;

3.2.1.2. define an action as (a) if condition is true, then permit the change of a state in a domain, (b) if condition is false, then forbid the change of a state in a domain;

3.2.2. in the case of a derivation axiom:

Note that a derivation axiom, which derives new information from the existing, can be a state axiom or a condition-state axiom.

3.2.2.1. in the case of a state axiom – transform an axiom to the corresponding action of an ECA rule. A condition is always true;

3.2.2.2. in the case of a condition-state axiom: (a) transform a condition to the corresponding condition of an ECA rule, and (b) transform a state to the corresponding action of an ECA rule.

4. End of the transformation.

The short description of the method is as follows. A user should choose an ontology with axioms to transform them to ECA rules. Each axiom is transformed to the corresponding ECA rule as follows. First, an event of an ECA rule is defined as *insert*, *update* or *delete*. Second, a condition and an action are defined according to the type of an axiom – consolidation or derivation. In the case of a consolidation axiom, it is transformed to the condition of an ECA rule. And an action is set as permit the change of a state, if condition is true, and forbid the change of a state, if condition is false. A *dynamic constraint* is obtained. In the case of a derivation axiom, the structure of an axiom is determined – a state or a condition-state axiom. In the case of a state axiom, it is transformed to the corresponding action of an ECA rule. A condition is always true. In the case of a condition-state axiom, a condition is transformed to the condition of an ECA rule and a state is transformed to the corresponding action of an ECA rule. A derivation or a reaction rule is obtained. *Note*, that in the simplest case all axioms can be transformed to the corresponding dynamic constraints. However, the semantics of axioms will be partially lost. As can be seen from the description of the method, not in all cases reaction rules are obtained. Therefore, the more detailed study of an ontology with axioms is necessary.

The method is independent of particular languages, which can be used for the definition of axioms and domain rules. The formal definition of the method using Z notation is presented in (Vasilecas *et al.* 2009).

4. A case study on ontology based implementation of qualification requirements for teachers

In this section the application of the proposed method for the implementation of qualification requirements for teachers is presented.

We choose the ontology development and management tool Protégé, to support our statement that ontology axioms can be transformed to information processing or even to executable rules. Ontology axioms are implemented in Protégé ontology by PAL constraints (Grosso and Crubezy 2008). Syntactical checking of defined axioms is available in Protégé also. Moreover, the free open source software can be installed locally and it is extensible. More ontology development and management tools are compared in (Kalibatiene and Vasilecas 2010).

A PAL constraint is a statement that holds on a certain number of variables, which range over a particular set of values. The language of PAL is a limited predicate logic extension of Protégé that supports the definition of such ranges and statements. The syntax of PAL is a variant of KIF. The main parts of a PAL constraint are presented as follows:

((`%3APAL-NAME` "[...]") (`%3APAL-RANGE` "[...]") (`%3APAL-STATEMENT` "[...]")),

where `%3APAL-NAME` holds a label for the constraint, `%3APAL-RANGE` holds the definition of local and global variables that appear in the statement, and `%3APAL-STATEMENT` holds the sentence of the constraint

The main part of the PAL constraint is the `%3APAL-STATEMENT`, which can be mapped to the corresponding rule. According to the structure, PAL statements are production rules, since they have clearly defined condition and a state. PAL constraints, like axioms, hold in an ontology constantly. Therefore, no information is provided about what should be done to implement a desirable state. The user should trigger PAL constraints manually, when it is necessary.

In this case study PAL constraints are transformed to SQL triggers. We choose SQL, since this language is one of the most popular languages in active database management systems (DBMS). The structure of an SQL trigger, which is presented below, corresponds to an ECA rule.

```
CREATE TRIGGER [ schema_name . ]trigger_name
ON { table | view }
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS { IF sql_statement  [ ; ] [ ...n ] }
```

where `[ schema_name . ]trigger_name` presents a name of a SQL trigger, `{ table | view }` presents a table or a view, to which this SQL trigger is attached, `FOR | AFTER | INSTEAD OF` denotes the execution time of an SQL trigger, `[ INSERT ]`

`[ , ] [ UPDATE ] [ , ] [ DELETE ]` denotes a triggering event, and `AS { IF sql_statement [ ; ] [ ...n ] }` denotes a body of an SQL trigger.

Figure 1 presents the mapping of a PAL constraint to an SQL trigger. This mapping is developed according to the proposed method in Section 3. According to the type, PAL constraint can be consolidation or derivation. According to the structure, PAL constraint can be state or condition-state. An SQL trigger fall into the set of ECA rules. Therefore, the proposed method is suitable for the transformation of PAL constraints to SQL triggers.

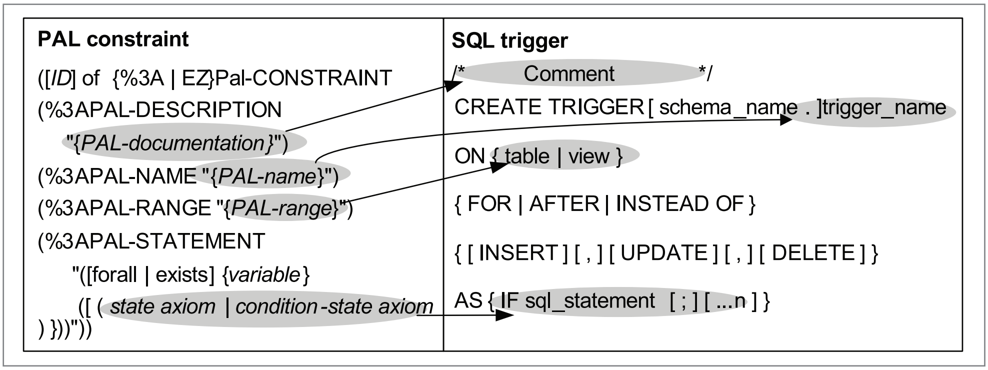

Fig. 1. The mapping of a PAL constraint to an SQL trigger (Vasilecas and Bugaite 2006)

A prototype of the *Axiom2SQL* plug-in was developed to carry out the experiment of automatic transformation of PAL constraints to SQL triggers. In this prototype it is necessary to specify a file, where SQL triggers will be stored. After the specification of a file all PAL constraints are transformed to SQL triggers automatically. Figure 2 presents the *Axiom2SQL* plug-in.

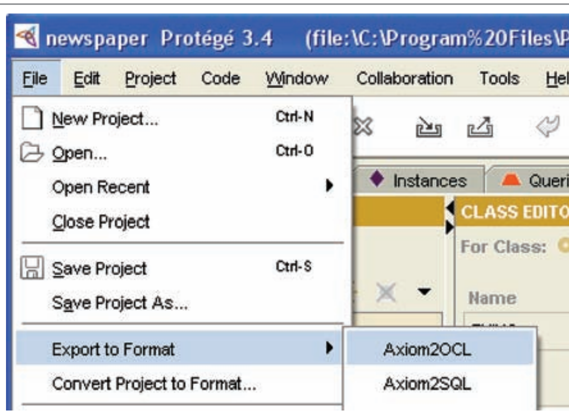

Fig. 2. The Axiom2SQL plug-in

At this moment plug-in is suitable for the implementation of dynamic constraints only. Therefore, in the future it should be extended to cover the transforming derivation axioms to derivation rules. an ontology of qualification requirements for teachers (QRT ontology) of our university is created to obtain an experiment of transforming PAL constraints to SQL triggers. A corresponding part of the QRT ontology is presented in figure 3.

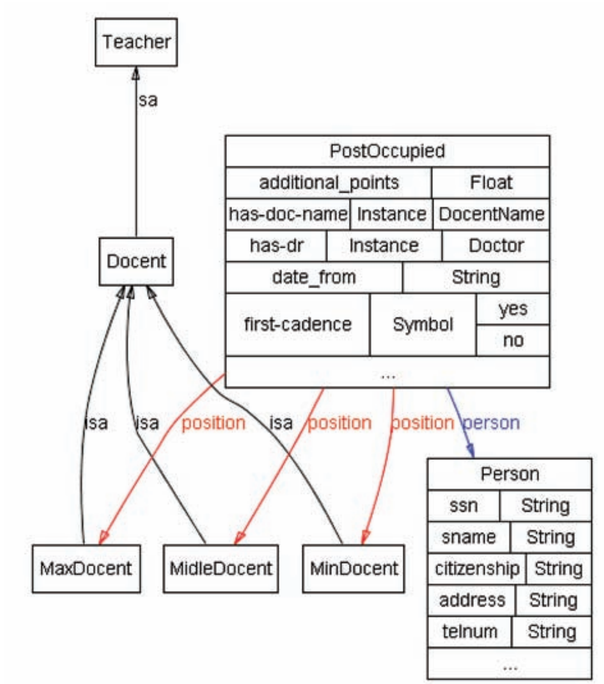

Fig. 3. The QRT ontology presented using Ontoviz plug-in (Sintek 2007)

The following four axioms are defined for the QRT ontology:

1. Every person has a unique social security number (Fig. 4).
2. Date_from must be before (less than) date_till.
3. A Person, who has senate-offering for the first cadence and has defended a doctor, or who has defended a doctor and a number of publications is less than 3, can occupy the post MinDocent for the first cadence (Fig. 5).

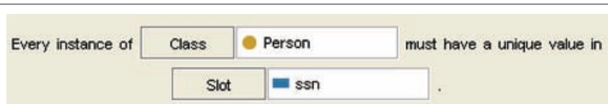

Fig. 4. The first axiom defined using EZPAL Tab plug-in (Hou *et al.* 2005)

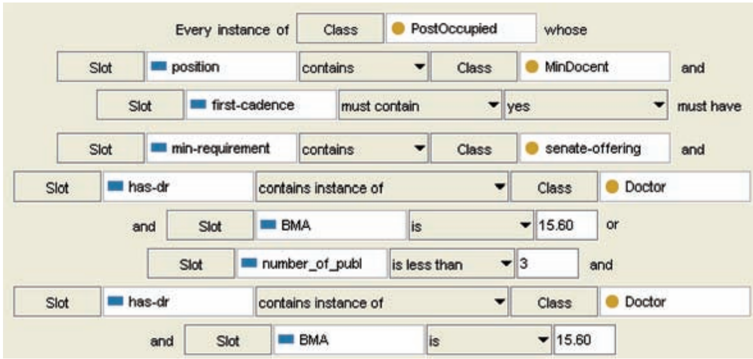

Fig. 5. The fifth axiom defined using EZPAL Tab plug-in

4. A Person, who has senate-offering for the first cadence and has defended a doctor, or who has defended a doctor and a number of publications is 3 and a number of additional point for other publications is greater than 15, can occupy the post MidleDocent for the first cadence (Fig. 6).

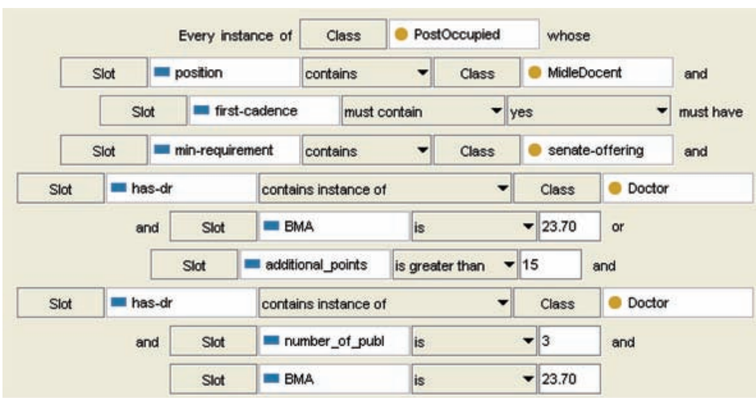

Fig. 6. The sixth axiom defined using EZPAL Tab plug-in

The following relational model was created to implement knowledge presented by the QRT ontology.

```
Person (ssn, fname, sname, birthdate, birthplace, gen-
der, citizenship)
PostOccupied (po, date_from, date_till, first_cadence,
has-dr, min-requirements, number_of_publ, additional_
points, sssn)
```

First axiom is implemented as a primary key in the relation Person. Second axiom is transformed to the SQL trigger as follows:

```
CREATE TRIGGER date-from-less-date-till
ON PostOccupied
{FOR | AFTER | INSTEAD OF}
{[DELETE] [,] [INSERT] [,] [UPDATE]}
AS
FOR EACH ROW
IF (date_from < date_till)
BEGIN
  COMMIT TRANSACTION
  PRINT ‘’Transaction is committed.’
END
ELSE
  RAISERROR (‘date_from mus be before date_till’)
  ROLLBACK TRANSACTION
END
```

Third and fourth axioms are transformed to the following SQL triggers.

```
CREATE TRIGGER minDocFirstCadence
ON PostOccupied
{FOR | AFTER | INSTEAD OF}
{[DELETE] [,] [INSERT] [,] [UPDATE]}
AS
FOR EACH ROW
IF (position = MinDocent AND first-cadence = ‘yes’ AND has-dr is not null) AND (min-requirement is not null) OR (number_of_publ > 3)
BEGIN
  COMMIT TRANSACTION
  PRINT ‘’Transaction is committed.’
END
ELSE
  RAISERROR (‘Rule (minDocFirstCadence) is violated.’)
  ROLLBACK TRANSACTION
END
CREATE TRIGGER midleDocFirstCadence
ON PostOccupied
{FOR | AFTER | INSTEAD OF}
{[DELETE] [,] [INSERT] [,] [UPDATE]}
AS
FOR EACH ROW
```

```
    IF (position = MidleDocent AND first-cadence = 'yes' AND
has-dr is not null) AND (min-requirement is not null) OR
(number_of_publ > 3 AND additional_points < 15)
    BEGIN
      COMMIT TRANSACTION
      PRINT `''Transaction is committed.'
    END
    ELSE
      RAISERROR ('Rule (midleDocFirstCadence) is violated.')
      ROLLBACK TRANSACTION
    END
```

Particular corrections should be made to SQL triggers obtained during the transformation. User should choose an activation time (`FOR | AFTER | INSTEAD OF`) and a triggering event (`[DELETE] [,] [INSERT] [,] [UPDATE]`) of an SQL trigger. The user should also link the generated SQL triggers with particular databases, since names of attributes or relations may vary because of implementation.

5. Conclusion and future works

The analysis of the related wors in the field of knowledge-based information systems development using the ontology shows that business rules are presented in the ontology by axioms. However, the topics of using ontology axioms for business rules modelling and consequently implementation in software systems are not investigated enougw.

According to the detailed analysis of ontology axioms and business rules, the method for transformation of ontology axioms to business rules is presented. Weesuggesdeusing of consolidation axioms for modelling of dynamic constraints, derivation axioms – for modelling of derivation rules, and epistemological axioms – for modelling of structuring of concepts.

In order to illustrate the presented approach and proof that it is implementable, ehe prototype of a software system Axiom2SQL was developet. The experiment obtained shows that Axiom2SQL can be used for the automatic transformation of ontology consolidation axioms to SQL triggers. It was concluded that the suggested method can be implemented and used for the automation of the ontology transformation to the components of software system of an information system.

References

Armonas, A.; Nemuraite, L. 2009. Using Attributes and Merging Algorithms for Transforming OCL Expressions to Code, *Information Technology and Control* 38(4): 283–293.

Badawy, M.; Richta, K. 2002. Deriving Triggers from UML/OCL Specification, in Kirikova, M. (Ed.). *Information Systems Development: Advances in Methodologies, Components and Management*. New York: Kluwer Academic / Plenum Publishers, 305–316.

Bizer, C. 2008. *DBpedia*. Available from Internet (December, 2008): <http://dbpedia.org/About>.

Boyd, L. 2001. CDM RuleFrame – The Business Rule Implementation Framework That Saves You Work, in *Proc. of ODTUG 2001*. Available from Internet (November, 2006): <http://www.dulcian.com/odtug_conference.htm>.

Ceponiene, L.; Nemuraite, L.; Vedrickas, G. 2009. Separation of Event and Constraint Rules in UML&OCL Models of Service Oriented Information Systems, *Information Technology and Control* 38(1): 29–37.

Culmone, R.; Rossi, G.; Merelli, E. 2002. An Ontology Similarity algorithm for BioAgent, in Ercolani,S.; Zamboni, M. A. (Eds.). *NETTAB Workshop on Agents and Bioinformtics*, *Bologna, 2002*. Available from Internet (March, 2007): <http://www.bioagent.net/WWWPublications/Download/NETTAB02P1.pdf>.

Demuth, B.; Hussmann, H.; Loecher, S. 2001. OCL as a Specification Language for Business Rules in Database Applications, in Gogolla, M.; Kobryn, C. (Eds.). *Proc. of the 4th International Conference on the Unified Modeling Language, Modeling Languages, Concepts, and Tools (UML 2001), Toronto, Canada, 2001*. London: Springer-Verlag, LNCS 2185: 104–117.

Dzemydiene, D.; Maskeliunas, S.; Dzemyda, I. 2008. Interoperability of information system components for monitoring of sewage and intelligent analysis of water resources, *Technological and Economic Development of Economy* 14(3): 260–278. doi:10.3846/1392-8619.2008.14.260-278

Falbo, R. A.; Menezes, C. S.; Rocha, A. R. C. 1998. A Systematic Approach for Building Ontologies, in Coelho, H. (Ed.). *Proc. of the 6th Ibero-American Conference on AI (IBERAMIA'98), Lisbon, Portugal, 1998*. Berlin / Heidelberg: Springer, LNAI 1484: 349–360.

Genesereth, M. R. 2006. *Knowledge Interchange Format (KIF)*. Available from Internet (October, 2006): <http://logic.stanford.edu/kif/kif.html>.

Grosso, W.; Crubezy, M. 2008. *The Protégé Axiom Language and Toolset ("PAL")*. Stanford Medical Informatics, Stanford University. Available from Internet (December, 2008): <http://protegewiki.stanford.edu/index.php/Protege_Axiom_Language_%28PAL%29_Tabs>.

Guarino, N. 1998. Formal Ontology and Information Systems, in *Proc. of FOIS'98*, *Trento, Italy, 1998*. Amsterdam: IOS Press, 3–15.

Gudas, S. 2009. Enterprise Knowledge Modelling: Domains and Aspects, *Technological and Economic Development of Economy* 15(2): 281–293. doi:10.3846/1392-8619.2009.15.281-293

Guizzardi, G.; Herre, H.; Wagner, G. 2002. On the General Ontological Foundations of Conceptual Modeling, in Spaccapietra, S., *et al.* (Eds.). *Proc. of 21th International Conference on Conceptual Modeling (ER 2002), Tampere, Finland, 2001*. Berlin: Springer-Verlag, LNCS 2503: 65–78.

Hassanpour, S.; O'Connor, M. J.; Das, A. K. 2009. Exploration of SWRL Rule Bases through Visualization, Paraphrasing, and Categorization of Rules, in *The International RuleML Symposium on Rule Interchange and Applications*, *Las Vegas, NV*. LNCS 5858: 246–261.

Hay, D. C. 2003. *Requirements Analysis: From Business Views to Architecture*. New Jersey: Prentice Hall PTR.

Herbst, H.; Knolmayer, G.; Myrach, T. and Schlesinger, M. 1994. The Specification of Business Rules: A Comparison of Selected Methodologies, in Verrijn-Stuart, A. A.; Olle, T. W. (Eds.). *Methods and Associated Tools for the Information System Life Cycle, 1994*. New York: Elsevier, 29–46.

Hou, C. S. J.; Noy, N. F.; Musen, M. A. 2005. EZPAL: Environment for composing constraint axioms by instantiating templates, *International Journal of Human-Computer Studies* 62(5): 578–596. doi:10.1016/j.ijhcs.2004.07.002

Kalibatiene, D.; Vasilecas, O. 2010. Ontology-based application for domain rules development, *Computer Science and Information Technologies* 38(4): 271–282.

Kaplinski, O. 2009. Information technology in the development of the Polish construction industry, *Technological and Economic Development of Economy* 15(3): 437–452. doi:10.3846/1392-8619.2009.15.437-452

Kapocius, K.; Butleris, R. 2006. Repository for Business Rules Based IS Requirements, *Informatica* 17(4): 503–518.

Lin, S.; Miller, L. L.; Tsai, H.-J.; Xu, J. 2001. Integrating a Heterogeneous Distributed Data Environment with a Database Specific Ontology, in Sha, E. H. M. (Ed.). *Proc. of the International Conference on Parallel and Distributed Systems 2001 (ICPADS'01), Kyongju City, Korea, 2001*. Washington: IEEE Computer Society Press, 430–435.

McGuinness, D.; Patel-Schneider, P. 1998. Usability issues in knowledge representation systems, in Mostow, J.; Rich, C. (Eds.). *Proc. of AAAI-98, Madison, Wisconsin, 1998*. Menlo Park: American Association for Artificial Intelligence, 608–614.

Nenortaite, J.; Butleris, R. 2009. Improving Business Rules Management through the Application of Adaptive Business Intelligence Technique, *Information Technology and Control* 38(1): 21–28.

Noy, N. F.; Fergerson, R. W.; Musen, M. A. 2000. The knowledge model of protege-2000: combining interoperability and flexibility, in Dieng, R.; Corby, O. (Eds.). *Proc. of the 12th International Conference in Knowledge Engineering and Knowledge Management (EKAWÕ00), Juan-les-Pins, France, 2000*. Berlin: Springer, LNAI, 1937: 17–32.

OMG. 2003. *UML 2.0 OCL Specification*. Available from Internet (September, 2008): <http://www.omg.org/docs/ptc/03-10-14.pdf>.

OMG. 2005. *Unified Modeling Language Specification*. Version 1.4.2. Available from Internet (September, 2008): <http://www.omg.org/cgi-bin/doc?formal/05-04-01>.

OMG. 2005a. *OntologyDefinition Metamodel*. Available from Internet (September, 2008): <http://www.omg.org/docs/ad/05-08-01.pdf>.

OMG. 2008. *Semantics of Business Vocabulary and Business Rules (SBVR)*. Version 1.0. Available from Internet (December, 2008): <http://www.omg.org/docs/formal/08-01-02.pdf>.

Ross, R. G. 1997. *The Business Rule Book. Classifying, Defining and Modeling Rules*. Houston: Business Rules Solutions Inc.

Ross, R. G. 2003. *Principles of the Business Rule Approach*. Addison Wesley.

Sintek, M. 2007. *OntoViz.* Available from Internet (March, 2010): <http://protegewiki.stanford.edu/index.php/OntoViz>.

Valatkaite, I.; Vasilecas, O. 2004. On Business Rules Approach to the Information Systems Development, in Linger, H. *et al.* (Eds.). *Proc. of the 12th International Conference on Information Systems Development (ISD'2003), Melbourne, Australia, 2003*. New York: Springer, 199–208.

Vasilecas, O.; Kalibatiene, D.; Guizzardi, G. 2009. Towards a Formal Method for Transforming Ontology Axioms to Application Domain Rules, *Information Technology and Control* 38(4): 271–282.

Vasilecas, O.; Bugaite, D. 2006. Ontology-based Information Systems Development: the Problem of Automation of Information Processing Rules, in Neuhold, E.; Yakhno, T. (Eds.). *Proc. of the Fourth International Conference Advances in Information Systems (ADVIS'2006), Izmir, Turkey, 2006*. Springer, LNCS, 4243: 187–196.

von Halle, B. 2002. *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. New York: John Wiley & Sons.

WordNet. 2006. Cognitive Science Laboratory, Princeton University. Available from Internet (December, 2008): <http://wordnet.princeton.edu/>.

Zacharias, V. 2008. *Technical Report: Development and Verification of Rule Based Systems – a Survey of Developers.* Available from Internet (May, 2009): <http://vzach.de/papers/2008_SurveyTechReport.pdf>.

ONTOLOGIJA GRINDŽIAMAS VERSLO TAISYKLIŲ MODELIAVIMAS

D. Kalibatienė, O. Vasilecas

Santrauka. Pastaruoju metu informacinės sistemos (IS) plačiai naudojamos įvairiose srityse verslo procesams valdyti. Verslo procesai ribojami įvairių taisyklių, apibrėžiamų įstatymais ir teisės aktais, verslo politikos ir tikslų. Todėl šios taisyklės turi būti efektyviai įgyvendintos ir valdomos verslą palaikančiose IS. Verslo taisyklių sąvoka plačiai pripažinta IS bendruomenėje kaip būdas išreikšti verslo ribojimus ir suvaržyti IS komponentus. Šiuo metu pasiūlyta nemažai metodų verslo taisyklėms išgauti iš dalykinės srities, apibrėžti, modeliuoti ir įgyvendinti. Tačiau nė vienas iš esamų metodų nėra pripažintas standartu, nes verslo taisyklių išgavimas, modeliavimas ir įgyvendinimas nėra trivialus uždavinys. Straipsnyje siūloma ontologija grindžiamas metodas verslo taisyklėms modeliuoti. Šis metodas leidžia automatizuoti verslo taisyklių įgyvendinimą, tam panaudojus ontologijoje apibrėžtas dalykinės srities žinias. Straipsnyje pateiktas pavyzdys, kai siūlomas metodas pritaikomas dėstytojų kvalifikaciniams reikalavimams įgyvendinti aktyvių duomenų bazių valdymo sistemų SQL trigeriais.

Reikšminiai žodžiai: verslo taisyklė, ontologija, aksioma, transformacija.

Diana KALIBATIENE. Dr Diana Kalibatiene is full time docent at the Information Systems Department, and junior researcher at the Information Systems Research Laboratory in Vilnius Gediminas Technical University. She participated in the High Technology Development Program Project "Business Rules Solutions for Information Systems Development (VeTIS)"". She is the member of the European Committee and Lithuanian Government supported SOCRATES/ERASMUS Thematic Network project "Doctoral Education in Computing" (ETN DEC) and "Teaching, Research, Innovation in Computing Education" (ETN TRICE). She is author of 30 papers and 1 book in the field of information systems development. Research interests: business rule and ontology based information systems development and conceptual modelling.

Olegas VASILECAS. Prof. Dr (HP). Olegas Vasilecas is full time professor at the information systems department, and senior researcher and head of Information Systems Research Laboratory in Vilnius Gediminas Technical University. He is author of more than 200 research papers and 5 books in the field of information systems development. His research interests: knowledge, including business rule and ontology, based information systems development. He delivered lectures in 7 European universities including London, Barcelona, Athens and Ljubljana. O. Vasilecas carried out an apprenticeship in Germany, Holland, China, and last time in Latvia and Slovenia universities. He supervised 6 successfully defended doctoral theses and now is supervising 5 doctoral students more. He was leader of many international and local projects. Last time he leaded "Business Rules Solutions for Information Systems Development (VeTIS)" project carried out under High Technology Development Program.