Ontology-Based Information Systems Development: The Problem of Automation of Information Processing Rules

Olegas Vasilecas and Diana Bugaite

Department of Information Systems, Vilnius Gediminas Technical University,
Sauletekio al. 11, LT-10223, Lithuania
Olegas.Vasilecas@fm.vtu.lt, diana@isl.vtu.lt

Abstract. The business rule approach is used in information systems to represent domain knowledge and to maintain rules systems efficiently in volatile business environment. A number of methods were proposed to develop rule models, but only few deal with reuse of knowledge acquired in the analysis of some particular domain and automatic implementation of rules. In this paper, a method for representing knowledge by ontology transformation into the rule model is described. The method is based on ontology transformation of axioms presented in a formal way into (semi-) formal information processing rules in the form of executable rules, like active DBMS triggers. The method is implemented into the developed prototype, which is described in the case study section.

Keywords: Ontology, ECA rule, SQL trigger, axiom.

1 Introduction

In the research of information systems development, the business rules approach has achieved a lot of attention and already has a strong motivation behind its application ([1,2,3]). A business rule is a directive, intended to govern, guide or influence business behavior, in support of business policy that has been formulated in response to an opportunity, threat, strength, or weakness [4]. Business rules are used in information systems to represent domain knowledge and to maintain rule systems efficiently in volatile business environment. A number of methods were proposed to develop rule models, but only few deal with reuse of knowledge acquired in the analysis of some particular domain and automatic implementation of rules. In computer science, ontologies are used to represent real-world domain knowledge. Therefore, knowledge represented by ontology can be used for generating rules. Moreover, ontology expressed in a formal way [5] can be transformed into rule model automatically.

In this paper, a method for representing knowledge by ontology transformation into the rule model is described. The method is based on transforming ontology axioms presented in a formal way into (semi-)formal information processing rules in the form of executable rules, like active DBMS triggers. The method is implemented into the developed prototype, which is described in the case study section.

T. Yakhno and E. Neuhold (Eds.): ADVIS 2006, LNCS 4243, pp. 187–196, 2006.

2 Related Work

A definition of a business rule (BR) depends on the context in which it is used. From the business system perspective, a BR is a statement that defines or constrains some aspects of a particular business. At the business system level, BRs are expressed in a declarative manner [6]. For example: *A customer could not buy more than credit limit permits.*

From the perspective of information systems (IS), a BR is a statement, which defines the major rules of information processing using a rule-based language [7]. Expressions of information-processing rules are very precise, e.g. terms used in expressions are taken from the particular data model. For example: *'Total Value' of an ORDER could not be greater than the 'Credit Limit' of a CUSTOMER* [8].

At the execution level (or software systems level), rules are statements that are transferred to the executable rules, like active DBMS triggers.

The more fundamental question in defining 'BR' can arise: 'What rules are BRs?' BRs are rules that are under business jurisdiction, e.g., the business can enact, revise and discontinue BRs as it sees fit [9].

Information-processing rules are used in ISs to process the required information correctly. These information-processing rules are derived from BRs, which are taken from the business system level. In practice, information-processing rules are implemented by executable rules. Information-processing rules should be expressed as ECA (*event-condition-action*) rules to be implemented by executable rules, like active DBMS SQL triggers.

Therefore it is necessary to determine and elicit rules from the application domain and develop ECA rules.

One of the possible ways to solve the defined problems is the use of the domain ontology.

The term 'ontology' is borrowed from philosophy, where Ontology means a systematic account of Existence. In computer science, the definition of ontology is rather confusing. By [10] all definitions of the term 'ontology' attempt to explain what an ontology is from three different aspects: the content of an ontology, the form of an ontology and the purpose of an ontology.

Gruber's definition of ontology, as a specification of a conceptualisation [11], is rather confusing. It explains the content of ontology, but does not explain what a conceptualisation is. According to Genesereth, a conceptualisation includes the objects and their relations which an agent presumes to exist in the world. The process of a conceptualisation is the process of mapping an object or a relation in the world to a representation in our mind [10].

Ontology defines the basic terms and their relationships comprising the vocabulary of an application domain and the axioms for constraining the relationships among terms [6]. This definition explains what an ontology looks like [10].

In the simplest case, an ontology describes a hierarchy of concepts related by particular relationships (like, is-a, part-of, etc.). In more sophisticated cases, constraints are added to restrict the value space of concepts and relationships. They, for example, express cardinality, possible length (like, maxLength, minLength) In most sophisticated cases, suitable axioms are added in order to express complex relationships between concepts and to constrain their intended interpretation [5].

Ontologies are being built today for many reasons. The reason of creating an ontology depends on a research field and an application area where it is going to be used. In this paper, ontology is used for its transformation into the rule model.

3 Transformation of Ontology Axioms into Business Rules

In the application domain or ontology, to which the BRs belong, they are not always expressed in terms of ECA rules. Some of these BRs have explicit or implicit condition and action parts. The missing condition can always be substituted with a default condition state as TRUE. Some BRs may have no explicit action since they can state what kind of transition from one data state to another is not admissible [12]. But the majority of these BRs do not define explicitly or implicitly the event. There are three possible ways to trigger rules:

- automatically trigger all rules every time when any related event occurs,
- trigger rules manually when somebody decides it is necessary,
- specify necessary events and link them to actual rules.

In this research, the third way was used for rules triggering, since the specification of the events and their linking to actual rules enable the system automatically react to the defined events and perform the defined operations, e.g. trigger rules automatically. Moreover, it is not necessary to execute all rules when some event occurs. Only related rules are executed.

Obviously, it is confusing to form information-processing rules of ECA form and consequently implement them by executable rules.

3.1 Formal Foundation of Ontology for Business Rules Elicitation

The mathematical models of ontology and business rules were analysed to determine the relationship between ontology and BRs.

From [13,14,15,16,17] it was determined that ontology could be expressed in the following way:

$$\Psi = \langle\{\Psi_i | i = 1, \ldots, k\}, A\rangle, \quad (1)$$

where Ψ_i is the ontology element which can be expressed by triplet:

$$\langle\nu_i, R'_I, I_i\rangle \quad \text{with } \nu_i \in V \wedge R'_i \in R' \wedge I_i \in I, \quad (2)$$

where $V = \{\nu_0, \nu_1, \ldots, \nu_n\}$ is a universal set of atomic terms, $R = \{r_0, r_1, \ldots, r_m\}$ is a universal set of relationships (e.g. is-a, synonym, related-to, part-of, etc.) between the terms and $I = \{I_i | i = 1, \ldots, n\}$ is a set of term definitions. A stands for the axioms expressing other relationships between terms and limiting their intended interpretation.

An axiom is a statement which is assumed to be true without proof [18]. Axioms define the state in which the domain should be. From the perspective of the ECA rule, axioms can have a clearly defined action and, sometimes, a condition under which the specific actions can be taken. When the state is changed, it is necessary to take an action. Events are not defined in axioms.

BRs-structural assertions [12] can be expressed as follows:

$$\langle \nu_i, \nu_j, c_i \rangle \quad \text{with } \nu_i, \nu_j \in V \wedge c_i \in C, \tag{3}$$

where ν_i and ν_j are terms used in structural assertions and c_i is a relationship-constraint, such as prerequisite (for example, *an order must have an order-data*), temporal (for example, *reservation precedes tour*), mutually-inclusive (for example, *to travel to a foreign country a VISA is required, based upon citizenship*), mutually-exclusive (for example, *a cruise cannot be listed as being sold out and have availability at the same time*) etc.

The analysis of formulas (1–3) allows us to state that terms and relationships expressing constraints which are used in structural assertions are adopted from sets of ontology terms and relationships. Therefore, we can assume that structural assertions are part of ontology.

The other part of rules is more complex. They consist of more than two terms and relationships between them. For example: *A customer must not place more than three rush orders charged to its credit account.*

These BRs are captured and fixed in the domain ontology by axioms (A).

Since BRs are captured in ontology by axioms and constraints of relationships among terms [6], ontology axioms (and ontology as a whole) represented in a formal way can be transformed into BRs (and into conceptual schema) automatically. Moreover, it facilitates BRs transformation into consequent information-processing and executable rules.

The general schema of axioms transformation into BRs is presented in Fig. 1. It is independent of implementation.

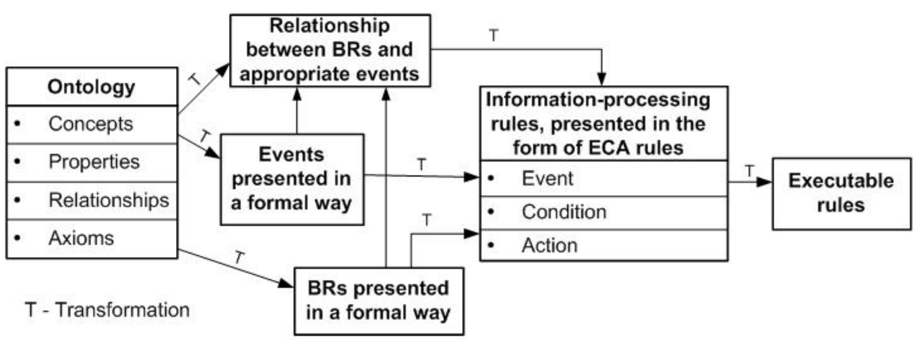

Fig. 1. Transformation of ontology axioms into ECA rules and consequent executable rules

Axioms do not stand alone in ontology. Since axioms define constraints on terms, terms are used to specify axioms. Therefore, these terms and their relationships should be transformed into (conceptual) schema in parallel with transformation of axioms into BRs. For the sake of simplicity, the schema of ontology axioms transformation into BRs is presented only in Fig. 1.

According to the examples (see [19,20,21,22,23]), there are two ways to define events in ontology. They are – to define necessary events by creating event ontology, which is related to other particular ontology/(-ies), or to define necessary events as a part of some other ontology. For the sake of simplicity, the second way for events definition was used, since in both cases, events are defined by terms used in ontology vocabulary.

Terms are used to link axioms with appropriate events, since ontology axioms and events are defined by terms used in ontology vocabulary (see Fig. 2).

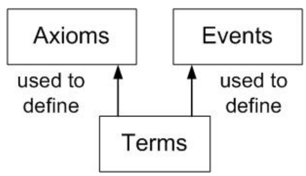

Fig. 2. Linking of ontology axioms and events

4 A Case Study of Ontology Axioms Transformation into Rule Model

The ontology for a particular business enterprise was created using Protégé-2000 ontology development tool to support the statement of the authors that ontology axioms and events can be transformed into information-processing rules and consequent executable rules. We chose Protégé-2000 to develop the ontology because it allows the open source software to be installed locally. A free version of the software provides all features and capabilities required for the present research as well as being user-friendly [24].

The axioms are implemented in Protégé-2000 ontology by the Protg Axiom Language (PAL) constraints. PAL is a superset of the first-order logic which is used for writing strong logical constraints [25].

The EZPal Tab plug-in is used to facilitate acquisition of PAL constraints without having to understand the language itself. Using a library of templates based on reusable patterns of previously encoded axioms, the interface allows users to compose constraints using a "fill-in-the-blanks" approach [26].

The definitions of the following axioms using the EZPal Tab are shown in Fig. 3 and Fig. 4.

Since PAL constraints are expressed in a formal way, it is reasonable to use this feature for their transformation into ADBMS SQL triggers.

The discount of a contract product depends on quantity of units of a contract product customer buys per time.
If quantity is greater then 19, discount is 3 %.

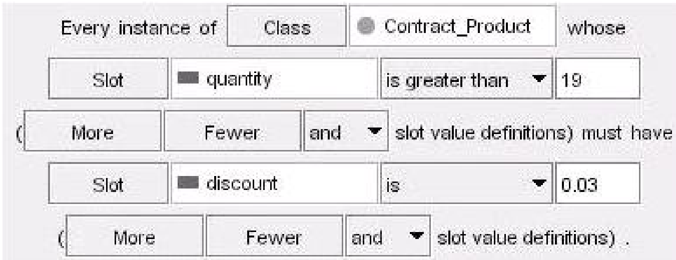

Fig. 3. An example of axiom creation using the EZPal templates

Every Contract must have a Contract_product.

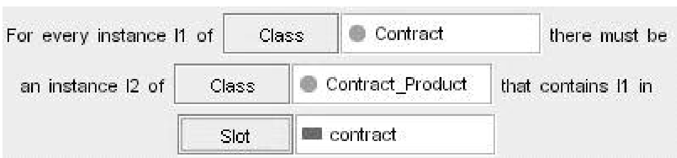

Fig. 4. An example of axiom creation using the EZPal templates

PAL constraints and SQL triggers were analysed in detail to automate constraints transformation into SQL triggers. The schema of automatic transformation is presented in Fig. 5.

The more detailed schema of PAL constraints transformation into SQL triggers is presented in Fig. 6. The main parts of PAL constraints (denoted by grey clouds) are transformed into the main parts of SQL triggers (denoted by grey clouds). 'PAL-documentation', 'PAL-name' and 'PAL-range' are transformed into SQL 'Comment', 'trigger_name', 'table | view' without significant changes. But 'if statement or condition' and 'action or possible state' should be properly processed and then transformed into 'sql_statement'.

Some fields of SQL trigger stay blank, since necessary information is lacking and cannot be taken from PAL constraints or Protégé-2000 ontology. For example, the time when trigger should be fired (FOR | AFTER | INSTEAD OF) should be specified manually.

The table or view on which the trigger is executed should be specified manually when more then one range in 'PAL-range' are defined. In other words, the trigger table or view should be chosen from the list of possible values manually. Specifying the owner name of the table or view is optional.

An event of a trigger (DELETE, INSERT, UPDATE) should be taken from events defined in ontology.

The relationships between axioms and events are determined by classes, used to define particular axioms and events. An example of the event is shown in Fig. 7.

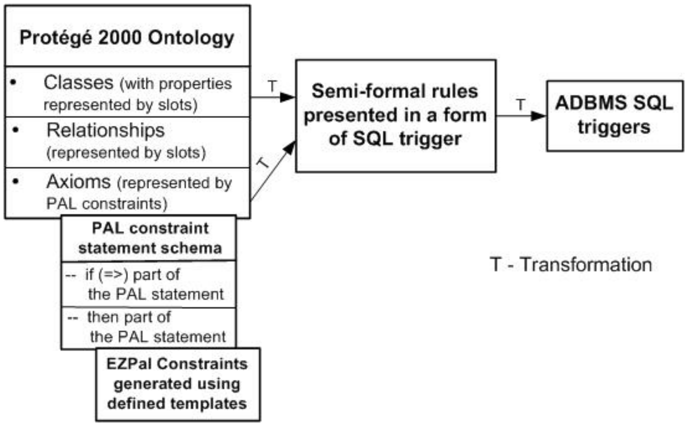

Fig. 5. The schema of automatic transformation of PAL constraints into SQL triggers

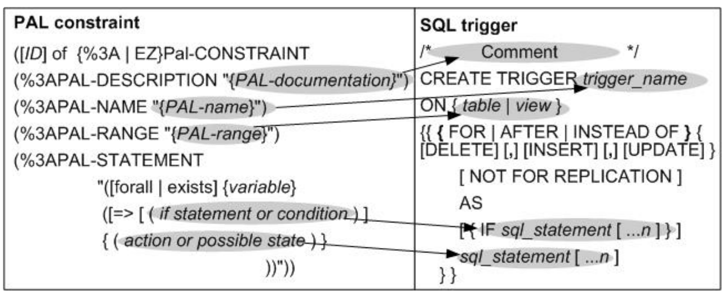

Fig. 6. The detailed schema of automatic transformation of ontology axioms into SQL triggers

For example, the event 'Making contract' (Fig. 7) and the axiom 'contract-must-have-contract-product' (Fig. 4) use the same class 'Contract' in their definitions. Moreover, the axiom 'contract-must-have-contract-product' is directly related with the axiom 'discount-3-percent' (Fig. 3), e.g., 'Contract_Product' is used in definitions of both axioms. Therefore, the following ECA rules can be generated:

When '*Making contract*', then '*Contract_product' should be defined.*
When '*Making contract*', if '*quantity' of a 'Contract_product'*
'is greater than 19',
then '*discount' of a 'Contract_product' 'is 3 %*'.

The prototype of necessary plug-in 'AxiomTransformation' was developed to carry out the experiment of ontology PAL constraints automatic transformation into semi-formal rules presented in the form of SQL trigger (see Fig. 8).

Fig. 7. An example of the event 'Making contract'

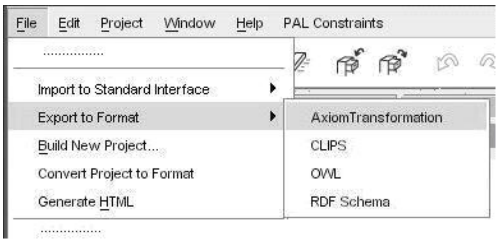

Fig. 8. The plug-in for Protégé-2000 ontology axioms transformation into semi-formal rules presented in a form of SQL trigger

It is necessary to specify a file, where only semi-formal rules will be saved. All axioms will be transformed into semi-formal rules automatically.

An example of the semi-formal rule presented in a form of SQL trigger is as follows:

```
/* Documentation */
/* The discount of a product depends on quantity of product units',
customer buys per time. If quantity is 10-19, discount is 3
CREATE TRIGGER discount-3-percent
ON
Contract_Product
{FOR | AFTER | INSTEAD OF}}
[DELETE] [,] [INSERT] [,] [UPDATE]
AS
(=> (> ('quantity' ?Contract_Product) 19)
(= ('discount' ?Contract_Product) 0.03)))
```

As mentioned above, some corrections are necessary for semi-formal rule to implement it by SQL trigger. User should also link the generated SQL triggers with particular DB, since some names of columns or tables can vary for some reasons. PAL-STATEMENT can be transformed into sql_statement only manually at the moment.

The next step of the research is extending the developed prototype, e.g., automation of PAL-STATEMENT transformation into sql_statement and events defined in ontology transformation into keywords (or events of the trigger) (DELETE, INSERT, UPDATE) that specify which data modification attempted against the table or view activate the trigger.

A full case study employing the proposed concepts and ideas of the proposed method is under development.

5 Conclusions and Future Work

The analysis of the related works on knowledge-based information system development using the domain ontology shows that the business rules are part of knowledge represented by ontology. Business rules are captured in ontology by axioms.

The method for ontology transformation into business rules, which are implemented by information-processing rules, was offered. We argue that the ontology axioms can be used to create a set of information-processing rules. They can be transformed into ECA rules and then to active DBMS triggers. Such transformation is possible, since ontology axioms can be mapped into active DBMS SQL triggers.

The prototype was developed and the experiment of ontology axioms automatic transformation into SQL triggers was carried out. The experiment shows that the suggested approach can be used to transform ontology axioms described in a formal way into SQL triggers. For this transformation, a suitable tool – Protégé-2000 – was chosen.

The next step in our research should be the refinement of the suggested method and the developed prototype. A full case study employing the proposed concepts and ideas is under development.

References

1. Business Rules Group: Defining Business Rules – What are they Really? Business Rules Group, 3rd Ed. (2000)
2. Morgan, T. Business Rules and Information Systems: Aligning IT with Business Goals, Addison-Wesley (2002)
3. Von Halle, B. Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley & Sons (2002)
4. Business Rules Group: The Business Motivation Model. Business Governance in a Volatile World, Business Rules Group (www.BusinessRulesGroup.org) (2005)
5. Guarino, N.: Formal Ontology and Information Systems. In Proc. of FOIS'98, Trento, Italy, 6–8 June, 1998. Amsterdam, IOS Press (1998) 3–15
6. Bugaite, D., Vasilecas, O.: Ontology-Based Elicitation of Business Rules. In A. G. Nilsson et all (eds.): Advances in Information Systems Development: Proc. of the ISD'2005. Springer, Sweden (2006) 795–806
7. Lebedys, E., Vasilecas, O.: Analysis of business rules modelling languages. In Proc. of the IT'2004, Lithuania, Technologija (2004) 487–494

8. Hay, D. C.: Requirement Analysis. From Business Views to Architecture. Prentice Hall PTR, New Jersey (2003)
9. Semantics of Business Vocabulary and Business Rules (SBVR). OMG (2005). (March, 2006): http://www.omg.org/docs/bei/05-08-01.pdf
10. Yang, X.: Ontologies and How to Build Them (2001). (March, 2006): http://www.ics.uci.edu/ xwy/publications/area-exam.ps
11. Gruber, T.: What is an Ontology? (March, 2004): http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
12. Valatkaite, I., Vasilecas, O.: On Business Rules Approach to the Information Systems Development. In H. Linger et all (eds.): Proc. of ISD'2003. Australia, Kluwer Academic/Plenum Publishers (2004) 199–208
13. Zacharias, V.: Kaon – towards a large scale semantic web. In Proc. of EC-Web 2002, LNCS, Springer (2002) 304–313
14. Goncalves, M. A., Watson, L. T., Fox, E. A.: Towards a Digital Library Theory: A Formal Digital Library Ontology. Virginia Polytechnic Institute and State University. (December, 2005): http://www.dcs.vein.hu/CIR/cikkek/MFIR_DLOntology4.pdf.
15. Culmone, R., Rossi, G., Merelli, E.: An Ontology Similarity algorithm for BioAgent in NETTAB Workshop on Agents and Bioinformatics, Bologna (2002). (December, 2005): http://www.bioagent.net/WWWPublications/Download/NETTAB02P1.pdf.
16. Hu, Z., Kruse, E., Draws, L.: Intelligent Binding in the Engineering of Automation Systems Using Ontology and Web Services. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 33, No.3 (2003) 403–412
17. Lin, S., Miller, L. L., Tsai, H.-J., Xu, J.: Integrating a Heterogeneous Distributed Data Environment with a Database Specific Ontology. The International Conference on Parallel and Distributed Systems (2001). (December, 2005): http://dg.statlab.iastate.edu/dg/papers_presentations/pdfs/lin_miller_et_al_2001_pdcs.pdf.
18. IMAGES. Glossary of Geometric and Mathematical Terms: Axiom (2004) (March, 2006): http://images.rbs.org/appendices/d_glossary_geometric.shtml.
19. Cilia, M., Bornhovd, C., Buchmann, A. P.: Event Handling for the Universal Enterprise. Information Technology and Management – Special Issue on Universal Enterprise Integration, Vol. 5, No.1 (2005) 123–148
20. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In Proc. of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services 2004, Boston (2004) (March, 2006): http://ebiquity.umbc.edu/get/a/publication/105.pdf
21. Event Ontology. KendraBase – server 3. (March, 2006): http://base4.kendra.org.uk/event_ontology
22. Ubiquity Event Ontology :: Classes and Properties. SchenaWeb (March, 2006): http://www.schemaweb.info/schema/SchemaInfo.aspx?id=114
23. Chen, H.: SOUPA Ontology 2004-06. semantic web in ubicomp (March, 2006): http://pervasive.semanticweb.org/soupa-2004-06.html
24. Jakkilinki, R., Sharda, N., Georgievski, M.: Developing an Ontology for Teaching Multimedia Design and Planning (September, 2005): http://sci.vu.edu.au/˜nalin/MUDPYOntologyPreprintV2.pdf
25. Grosso, W.: The Protg Axiom Language and Toolset ("PAL") (2002) (September, 2005): http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.html
26. Hou, J.: EZPal Tab. Stanford University (2005) (March, 2006): http://protege.stanford.edu/plugins/ezpal/