Framework on application domain ontology transformation into set of business rules

Diana Bugaite, Olegas Vasilecas

Abstract: *Research on ontology is becoming increasingly widespread in the computer science community, and its importance is being recognized in a multiplicity of research fields and application areas, including and knowledge engineering. Ontology represents the real-world domain knowledge, which part is business rules. We analyse ontology and business rule concepts and define how business rules are related with domain ontology. The main purpose of this paper is to show how domain ontology can be used for eliciting of business rules.*

Key words: *Formal ontology, business rules model, ECA rules.*

INTRODUCTION

Research on ontology is becoming increasingly widespread in the computer science community, and its importance is being recognized in a multiplicity of research fields and application areas, including and knowledge engineering, as ontology represents the real-world domain knowledge.

The fast changing requirements are the main problem of creating and/or modifying applications. Most of these requirements are in the form of or are related to business rules.

Business rules (BR) control and influence the behaviour of systems in the organization. BR are derived from business policies, and these in turn are the direct implementation of business goals and objectives. The process of finding out what the set of rules applies to a given business situation often involves an open-ended search through multiple sources [1]. The consensus from all the business stakeholders should be obtained on what the rules should be. Therefore, it is vital to find the rules and ensure that the rules are appropriate. As the business changes, the set of rules must be properly maintained and adapted to the new conditions. BR capturing, documenting and retaining prevent the loss of knowledge, when people leave an enterprise [2].

As BR are the part of the domain knowledge, ontology can be used to form a set of BR.

The objective of this paper is to show, how domain ontology can be used to elicit BR.

RELATED WORKS ON BUSINESS RULES AND ONTOLOGY

As this paper is addressed to an interdisciplinary audience, it is advisable to pay attention to some terms to be used.

Definition of a BR depends on context we use. From a business perspective, a BR is a statement that defines or constrains some aspects of the business; it is intended to assert the business structure, or to control or influence the behaviour of the business [3, 4]. The final set of BR must comprise only atomic BR. The atomic BR are such BR that cannot be broken down or decomposed into more detailed BR, because there would be loss of important information about the business. From perspective of information systems (IS), a BR is a statement, which constrains business aspect, defines the business structure and controls business processes [3].

At the business system level, the BR are the statements that express business policies in a declarative manner. At the IS level, the BR are the statements that define rules of information processing using a rule-based language [5].

We will use the following definition of the BR in our paper:

Business rule is a statement, which constrains or influences some business aspects, defines the business structure, and controls business processes.

Different taxonomies of business rules are presented in [2]. But from an implementation perspective all business rules can be classified into structural assertions (like terms, facts, integrity constraints) and dynamic assertions (like dynamic constraints, derivation rules and reaction rules) [2]. We chose this taxonomy, because we are concentrated on the BR, which enable to implement the active behaviour of modern information systems. They are dynamic assertions. Dynamic assertions smoothly map to the ECA paradigm (when *event* occurs, if *condition* is true, then *action*).

There are the following BR eliciting problems:

1. Business representatives don't use the expert systems or program languages to express BR. The business representatives express BR in "businesses speak". Therefore, it is difficult to identify BR from the "speak" and documents used in business [4].
2. Because of double meaning of some terms used in business, it is necessary to use not only terms, but and concepts, e.g. it is necessary to clearly understand and define terms used in business [6]. The simple example of the BR can be: *An order must be placed by a customer*. Terms "*order*" and "*customer*" have to be clearly and unambiguously defined by a sentence [1].
3. It is difficult to form consistent, integral, correct and complete set of BR [1, 6].

As BR are the part of the domain knowledge, ontology can be used to elicit BR.

The term "ontology" is borrowed from philosophy, where Ontology is a systematic account of Existence. The most popular is T. Gruber's definition of the ontology, where in the context of knowledge sharing, the term "ontology" means a *specification of a conceptualisation* [7].

The subject of ontology is the study of the categories of things that exist or may exist in some domain [8].

Every IS has its own ontology, since it ascribes meaning to the symbols used according to the particular view of the world [9]. The role of ontology is to provide a comprehensive set of terms, definitions, relationships and constraints for domain, e.g. the domain ontology is used as domain model in IS engineering and development [10, 11].

We will use the following ontology definition:

Ontology defines the basic terms and their relationships comprising the vocabulary of an application domain and the axioms for constraining relationships among terms.

Classification of ontologies were analysed to understand, which ontologies are used in IS development process.

In IS context, there is usually used classification of ontologies according to their level of generalisation. At the top level there is a *general ontology*, which is independent of a particular problem or domain. It describes very general terms like space, time, object, event, action, etc. In the middle level there are *domain or task ontologies*, which describe the vocabulary related to a generic domain (like medicine, automobiles, or business) or a generic task or activity (like diagnosing or selling), by specializing the terms introduced in the general ontology. And in the lowest level there are *highly specialised domain ontologies* (or *application domain ontologies)*, which specify domain or task ontologies [9, 11, 12]. Such ontologies are used for development of single applications, so-called micro-worlds. The principal advantage of such ontology is easiness of design and implementation. However, it is difficult to reuse the highly specialised domain ontology in other micro-worlds. So, enterprise engineering requires even more generic shared ontologies that can support applications across many domains, e.g. across all areas of enterprise's business [11].

Domain ontology, which specifies conceptualisations specific to domain, is directly related to IS development [10].

An IS is ontology-driven, if the ontology plays the central role in the IS life cycle [13].

An IS consists of such main components:

- Application programs,
- Information resources like databases, knowledge bases and,
- User interfaces.

These components are integrated so as to accomplish a business purpose.

Using an ontology approach in an IS development, the ontology becomes a separate component of an information system. This component can be used by other IS components for different purposes [9, 13]. The impact the ontology has on IS can be twofold [9]: at *development time* (*for* an IS) and at *run time* (*within* an IS). We are interested in the first one in this paper.

First, IS development using the ontology approach depends on what the ontology we have: domain or generic ontology. If we have the domain ontology, then the result of development phase will be application ontology. It enables the developer to reuse knowledge instead of software and share application domain knowledge using a common vocabulary across heterogeneous software platforms. If we have the generic ontology, we can develop highly specialised domain ontology without necessarily having domain ontology [9].

Second, each of the components of the IS can use the ontology in its own specific way [9]:

- Database component:
  - Ontology can be used in the requirement analysis.
  - Ontology can suggest missing entities and relationships among them for the application domain [10].
  - Ontology can be compared with database conceptual model. It can be represented as a computer-processable ontology and from there mapped to concrete target platforms [9].
  - Such conceptual model (computer-processable ontology) can be used in mapping heterogeneous conceptual schemes (information integration) [9, 10].
- User interface component – semantic information embodied in ontology can be used to generate form-based interfaces. It was successfully used in the Protégé Project [10].
- Application program component – ontology can be used to generate static (type or class declarations) and procedural (like, e.g. business rules) parts of a program [9, 10].

In the related works there are not explicitly presented how ontology is related to BR and methods to generate a set of BR is not presented as well.

We make an assumption that using an ontology in IS development helps to solve the BR eliciting problems defined above. E.g. as the domain ontology represents the real-world domain knowledge and BR are specific part of all domain knowledge, ontology can be used for BR eliciting. Domain ontology clearly and unambiguously defines the basic terms and their relationships, which are used in BR.

For this it is necessary to investigate, how the domain ontology is related with BR.

USING ONTOLOGY FOR BUSINESS RULES ELICITING

The mathematical models of ontology and BR need to be created to investigate the relationship between ontology and BR.

The number of ontology definitions and models [14, 15, 16, 17, 18] were analysed and it was determined that ontology can be expressed by the expression:

$$\Psi = < \{\Psi_i \mid i = 1,...,k\}, A > , \quad (1)$$

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

$$< v_i, R'_i, I_i > \text{ with } v_i \in V \wedge R'_i \in R' \wedge I_i \in I , \quad (2)$$

where $V = \{v_0, v_1,..., v_n\}$ is the universal set of atomic terms, $R = \{r_0, r_1,..., r_m\}$ is the universal set of relationships (like *is-a*, *synonym*, *related-to*, *part-of*, etc.) between the terms and $I = \{I_i \mid i = 1,..., n\}$ is the set of term definitions. *A* is the axioms, which express other relationships between terms and constrain their intended interpretation [9] (see below).

Structural assertions can be expressed in the following way:

$$< v_i, v_j, c_i > \text{ with } v_i, v_j \in V \wedge c_i \in C , \quad (3)$$

where v_i and v_j are terms used in structural assertions and c_i is a relationship-constraint, like *must*, *must not*, *should*, *should not* or *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*) relationships, etc.

The analysis of formulas (1-3) let us state that terms and relationships used in structural assertions are adopted from the sets of ontology terms and relationships. Therefore, we can assume that structural assertions are the part of ontology.

Then, if $T_1(t_1,..., t_n)$ is a set of structural assertions, which are consist of two terms and relationship among terms, the ontology element Ψ_i can be expressed by the twain:

$$< t_i, I_i > . \quad (4)$$

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

These BR are captured in the domain ontology by *axioms* (*A*).

The axioms define constraints and BR on terms and other horizontal relationships among them. The theory of axioms is based on situation calculus and predicate calculus for representing the dynamically changing world [13]. Situation theory views the domain as having a state. When the state is changed, there is necessary to take an action. The predicate theory defines conditions on which specific actions can be taken.

The simple example of a formal representation of axiom states that, if there is a product, its demand should exist [13]: *Exist (demand/product)*.

The axioms represent the intension of concept types and relation types and, generally speaking, knowledge which is not strictly terminological [19].

In general the framework of transformation of ontology axioms into ECA rules and active DBMS SQL triggers is shown below (Figure 1).

Ontology axioms (and the whole ontology) represented in a formal way can be transformed into the BR, the BR can be transformed into the information processing rules (like ECA) and the information processing rules can be transformed into the executable rules (like SQL triggers in active DBMS).

The ontology of the specific business enterprise was created using Protégé-2000 ontology development tool to support our preposition that BR can be elicited from the domain ontology. We chose Protégé-2000 to develop our ontology, because it is possible to install locally the open source software; the free version of the software provides all required for our research features and capabilities; it maintains writing of constraints; it is Java based, etc. [20]. The axioms are implemented in Protégé-2000 ontology by the

Protégé Axiom Language (PAL) constraints. PAL is a superset of the first-order logic, which is used for writing strong logical constraints [21].

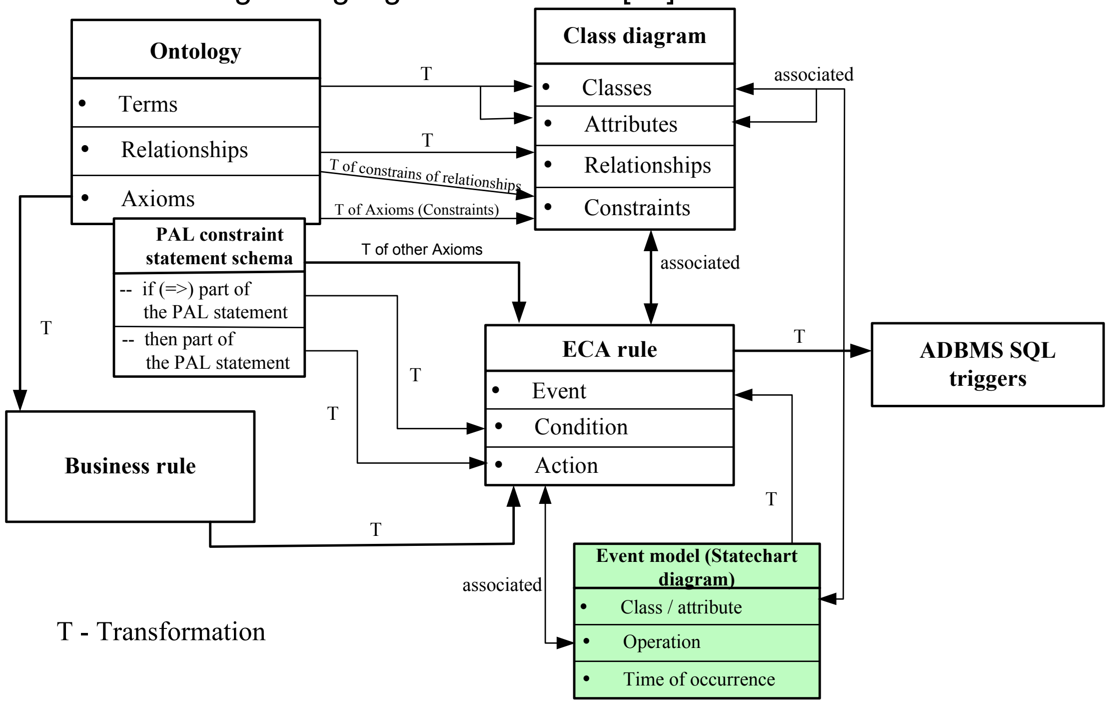

Figure 1: Ontology transformation into ECA rules and active DBMS SQL triggers.

Our analysis shows that ontology axioms, which can be mapped to the business rules and consequently to the ECA rules or directly to the ECA rules, *have* clearly defined *action* and sometimes *condition* parts, and *haven't* defined *event* part. All ontology axioms define the state, in which domain should be, none what should be done to implement desirable state. Therefore, it isn't obvious, when a rule has been triggered. So, it is necessary to extend ontology axioms by event model, which will be related with triggering of rules (Figure 1).

CONCLUSIONS

The analysis of related works on knowledge based IS development using the domain ontology shows that the BR are the part of knowledge represented by the ontology. BR are captured in ontology by axioms and relationships-constraints between terms.

Our analysis of mathematical models of ontology and BR shows that these models are compatible. Therefore, domain ontology can be used to create a set of BR.

In our paper we propose a framework, which can be used for the domain ontology axioms transformation into the ECA rules and then into the active DBMS triggers.

The example developed using Protégé-2000 shows, that such framework can be implemented. Event model should be added to the system to automate transformation.

REFERENCES

[1] R. G. Ross. Principles of the Business Rule Approach. Addison Wesley. 2003. p. 352.

[2] I. Valatkaite, O. Vasilecas. On Business Rules Approach to the Information Systems Development. In H. Linger at al (Eds.). Proc. of Twelfth International Conference on Information Systems Development. Constructing the Infrastructure for the Knowledge Economy. Melbourne, Australia, Kluwer Academic/Plenum Publishers, 2004, pp. 199-208.

[3] J. A. Zachman, J. F. Sowa. Extending and formalizing the framework for information systems architecture IBM Systems Journal, 1992, Vol. 31, no. 3, IBM Publication, p. 590-616.

[4] Why Use Rules? Role Machines Corporation. 2003. URL: http://www.rulemachines.com/VRS/whyrules.htm.
[5] O. Perrin, C. Godart. An approach to implement contracts as trusted intermediaries. Proceedings of the First International Workshop on Electronic Contracting (WEC'04), 2004, pp. 71-78.
[6] D. S. Appleton. Business Rules: The Missing Link. URL: http://www.defenselink.mil/nii/bpr/bprcd/5001.htm. [2004.03.24].
[7] T. Gruber. What is an Ontology? URL: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html. [2004.03.20].
[8] J. F. Sowa. Knowledge Representation: Logical, Philosophical, and Computational Foundation. Brooks/Cole, Thomson Learning, Pacific Grove, CA. 2000.
[9] N. Guarino. Formal Ontology and Information Systems. In Proc. of FOIS'98, Trento, Italy, 6-8 June, 1998. Amsterdam, IOS Press, pp. 3-15.
[10] V. Sugumaran, V. C. Storey. Ontologies for conceptual modeling: their creation, use, and management. Data Knowl. Eng. 42(3), 2002, p. 251-271.
[11] A. Caplinskas, A. Lupeikiene, O. Vasilecas. The Role of Ontologies in Reusing Domain and Enterprise Engineering Assets. INFORMATICA, 2003, Vol. 14, No. 4, pp. 455-470.
[12] D. Ciuksys, A. Caplinskas. Overview of Ontology-Based Approach to Domain Engineering. Information sciences. Vilnius University Publishing House, 2003. Vol. 26. p. 94-97. (In Lithuanian).
[13] C. Chandra, A. Tumanyan. Supply Chain System Analysis and Modeling Using Ontology Engineering. In Proc. of the Tenth Americas Conference on Information Systems, New York, August 2004, p. 4309-4319.
[14] V. Zacharias. Kaon - towards a large scale semantic web. In Proceedings of EC-Web 2002, LNCS, pages 304–313. Springer, 2002.
[15] M. A. Goncalves, L. T. Watson, E. A. Fox. Towards a Digital Library Theory: A Formal Digital Library Ontology. Virginia Polytechnic Institute and State University. URL: http://www.dcs.vein.hu/CIR/cikkek/MFIR_DLOntology4.pdf. [2004.12.06].
[16] R. Culmone, G. Rossi, E. Merelli. An Ontology Similarity algorithm for BioAgent in NETTAB Workshop on Agents nd Bioinformtics, Bologna, 2002. URL: http://www.bioagent.net/WWWPublications/Download/NETTAB02P1.pdf.
[17] Z. Hu, E. Kruse, L. Draws. Intelligent Binding in the Engineering of Automation Systems Using Ontology and Web Services. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviewes, Vol. 33, No.3, August, 2003. pp. 403-412.
[18] S. Lin, L. L. Miller, H.-J. Tsai, J. Xu. Integrating a Heterogeneous Distributed Data Environment with a Database Specific Ontology. The International Conference on Parallel and Distributed Systems, 2001. URL: http://dg.statlab.iastate.edu/dg/papers_presentations/pdfs/lin_miller_et_al_2001_pdcs.pdf [2004.12.06].
[19] F. Fürst, M. Leclère, F. Trichet. TooCoM : a Tool to Operationalize an Ontology with the Conceptual Graphs Model. URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-87/EON2003_Fuerst.pdf. [2005.03.05].
[20] R. Jakkilinki, N. Sharda, M. Georgievski. Developing an Ontology for Teaching Multimedia Design and Planning. URL: http://sci.vu.edu.au/~nalin/MUDPYOntologyPreprintV2.pdf. [2005.01.05].
[21] Monica Crubézy. The Protégé Axiom Language and Toolset ("PAL"). 2002. URL: http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.html.

ABOUT THE AUTHORS

Prof. Olegas Vasilecas, Dr, Department of Information Systems, Vilnius Gediminas Technical University, Phone: +37052744859, E-mail: olegas@fm.vtu.lt.

Diana Bugaite, PhD student, Department of Information Systems, Vilnius Gediminas Technical University, E-mail: diana@isl.vtu.lt.