On Approach for Enterprise Ontology Transformation into Conceptual Model

Olegas Vasilecas, Diana Bugaite, Justas Trinkunas

Abstract: *Conceptual models of information systems are used to capture the meaning of an application domain as perceived by someone. The important requirement for developing conceptual models is to reduce efforts, costs and time. This requirement can be implemented by the explicit use of enterprise knowledge for automatic generation of conceptual model. A number of methods were proposed to develop conceptual models, but only few deals with knowledge use. In this paper we present an approach for knowledge represented by ontology automatic transformation into conceptual model, which can be transformed into executable specification. The meta-models of ontology and conceptual model are analysed and the method of automatic transformation is proposed. The developed prototype, which realise proposed method, is described in the case study section.*

Key words: *conceptual model, ontology, meta-model.*

INTRODUCTION

Conceptual modelling methodologies are well developed and have proven to be quite successful for building information systems in a graphical way at the conceptual level.

But there are the following main problems concerning conceptual modelling. Firstly, to employ conceptual modelling constructs effectively, their meanings have to be defined rigorously. Often, however, rigorous definitions of these constructs are missing. [1] Secondary, in general, most conceptual schemas are developed from scratch, which means wasting previous efforts and time [2]. Thirdly, domain knowledge is not used for conceptual modelling.

Because conceptual models are intended to capture knowledge about a real-world domain, the meaning of modelling constructs should be sought in models of reality. Accordingly, ontology, which is the branch of philosophy dealing with models of reality, can be used to analyse the meaning of common conceptual modelling constructs.

The objective of this paper is to investigate how ontology can be used to generate a conceptual model.

RELATED WORK

Conceptual modelling (or semantic modelling) focuses on capturing and representing certain aspects of human perceptions of the real-world so that these aspects can be incorporated into an information system [1]. Most conceptual modelling approaches are concerned with essential concepts, associations among concepts and constraints of a domain [3].

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 [4] 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. The three aspects of ontology are described as follows.

Gruber defined ontology as a specification of a conceptualisation [5]. Gruber's definition explains the content of ontology, but it is confusing because it 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. [4]

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 [4].

As was mentioned before, in this paper, the emphasis is placed on the usage of ontology for the generation of a conceptual model.

Conceptual data models and ontologies are quite similar, as both consist of concepts, relationships between them and rules (in ontology – axioms), which are added in order to express other relationships between concepts and to constrain their intended interpretation. In conceptual modelling they are commonly called 'constraints'. [7].

Main disparities between ontologies and conceptual model are presented in [7]. They are not discussed in this paper.

But a one-to-one mapping does not exist between ontology and conceptual model [1] in all cases, since ontology of a domain is richer then a conceptual model of the same domain. There is no possibility and sometimes necessity to present some ontological aspects in a conceptual model. For example, there is no possibility to present synonym and antonym relationships in conceptual model. But that knowledge about domain is necessary to understand it completely. Also it is important to note that several different conceptual models can be build from the same domain ontology. It depends on what we need to present in a conceptual model. Therefore, it is important to develop necessary rules for ontology transformation into conceptual model. Some general rules are proposed in [1].

Without the ability to perform automatic model transformations, every existing model must be developed and understood separately, and/or has to be converted manually between the various modelling formalisms. This often requires as much effort as recreating the models from scratch, in another modelling language. However, when automatic model transformations are used, the mapping between the different concepts has to be developed only once for a pair of meta-models, not for each model instance. [8]

To define the mapping between two conceptually different models requires a common basis that describes both the source and target domains of the transformation, and the transformation vocabulary. This common basis in this case is the meta-model [8].

META-MODEL BASED ONTOLOGY TRANSFORMATION INTO A CONCEPTUAL MODEL

The schema of models transformation from [8] was adapted to the ontology transformation into conceptual model.

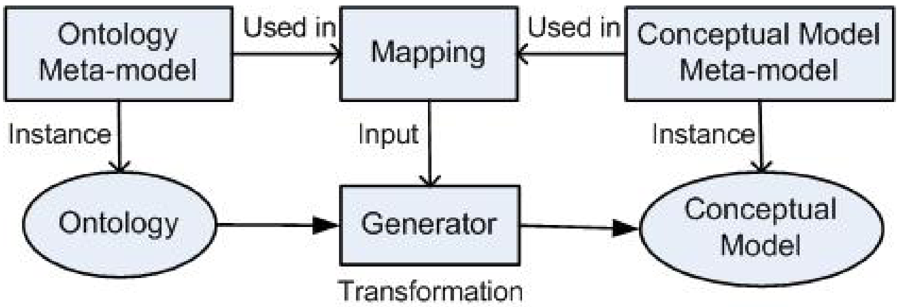

Figure 1: Meta-model based ontology transformation into a conceptual model.

Meta-models of ontology and conceptual model should be analysed and mapped to perform automatic ontology transformation into conceptual model.

A number of ontology meta-models (OWL Meta-model [9], RDF Schema (RDFS) Meta-model [9], ontological foundation from [1], meta-model from [10]) and the conceptual model meta-models (ER meta-model [9], ORM meta-model [11]) were analysed and it was determined that the summarised transformation of ontology into conceptual model is the following:

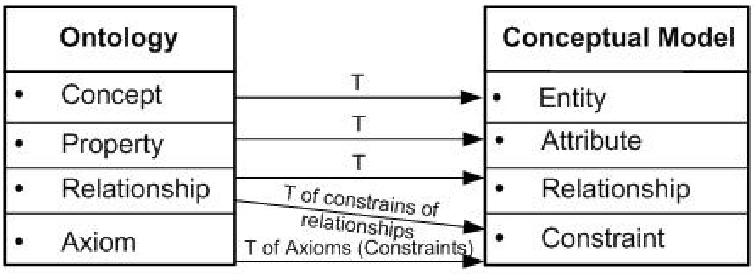

Figure 2: Meta-model based ontology transformation into a conceptual model.

In Figure 2, concepts that serve to designate a category or class of entities, events or relations are transformed into entities. Properties that designate the characteristics of concepts are transformed into attributes of particular entities. Relationships between concepts are transformed into relationships between entities. Ontology axioms are transformed into constraints.

The analysis of ontology meta-models shows that different authors name axioms in different way. For example, in [1] some properties of things (in our case, concepts) are called predicates. However, they are speaking about the same thing – axioms.

In the simplest case, an ontology describes a hierarchy of concepts related by relationships (like, is-a, part-of, e.g.). 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 [12].

Meta-models of Protégé-2000 ontology [13] and ER model [9] were chosen for detailed analysis.

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. It also maintains multiple inheritances, provides exhaustive decomposition, disjoint decomposition and constraints writing as well as being Java-based [14].

Concepts are implemented in Protégé-2000 by classes, properties and relationships – by slots, axioms – by the Protégé Axiom Language (PAL) constraints. PAL is a superset of the first-order logic which is used for writing strong logical constraints [15].

The ER diagram has been widely used as a means for describing conceptual or logical models [9].

The summary of the Protégé-2000 ontology into ER model mapping is the following:

Table 1: Protégé-2000 Ontology mapping into ER model.

Source: Protégé-2000 Ontology	**Target: ER Model**
Class	Entity
Slot 1. Documentation 2. Value Types: ▪ Any ▪ Boolean ▪ Class ▪ Float ▪ Instance ▪ Integer ▪ String	Attribute 1. Comment 2. Data Types: ▪ Other ▪ Boolean ▪ Entity ▪ Float ▪ Foreign Key and/or relationship with appropriate Entity ▪ Integer ▪ Variable characters

▪ Symbol 3. Required 4. Minimum 5. Maximum 6. Default Values	▪ Domain or List of Values 3. Mandatory 4. Values: Minimum 5. Values: Maximum 6. Values: Default
Directed-Binary-Relation	Inheritance
Constraint	Entity Constraint

The elements of the Protégé-2000 ontology are listed in the left column. They are transformed into the ER model elements listed in the right column.

There are the following type constraints in ER model meta-model:

- Domain Constraints [9]. Domain constraints are used to restrict the value space of data types. A value space is the set of values for a given data type. Some examples of constraints are: length, maxLength, minLength, e.g.
- Entity Constraints [9]. Entity constraints are used to restrict the instances of entities. All instances of an entity must satisfy its entity constraints. Entity constraints can be defined using allowed languages (English, OCL and XML). But nothing is written about possible constraint types.

Power Designer was chosen to implement ER model, since it is suitable for this research. The software provides all features and capabilities for the conceptual model development, its automatic transformation into the physical model and consequently into a code, which can be executed on MS SQL, Oracle, DB2 and other modern DBMSs.

Power Designer supports all ER model capabilities mentioned above (Table 1), except complex constraints. Consequently Protégé-2000 constraints (or axioms) should be transformed, saved and linked with a particular ER model in another way. The topic of ontology axioms transformation into constraints and business rules are not discussed here in more details (see [6, 16]).

CASE STUDY

The ontology for a particular business enterprise was created using Protégé-2000 to support the statement of the authors that ontology can be transformed into a conceptual model.

The result of the developed ontology transformation into the conceptual model is presented in Figure 3.

The one-to-one mapping is used, since ontology concepts of the enterprise match entities in the ER model. Such one-to-one mapping can be used in case of ontology represent particular domain. In general ontology concepts are abstract and can't be transformed into ER model entities directly. Therefore the one-to-one mapping is impossible.

Enterprise ontology created with Protégé-2000 is presented in the left side of the Figure 3. All classes from the ontology class hierarchy are transformed into ER model entities presented in the right side of the Figure 3. Systems-class, Property and Template in the ontology are not transformed into the ER model, because those classes do not represent any domain knowledge.

The slots of the classes were transformed into the attributes of the conceptual model also (Figure 3).

The next step in our research is extending the developed prototype and the full case study employing the ideas of the proposed approach.

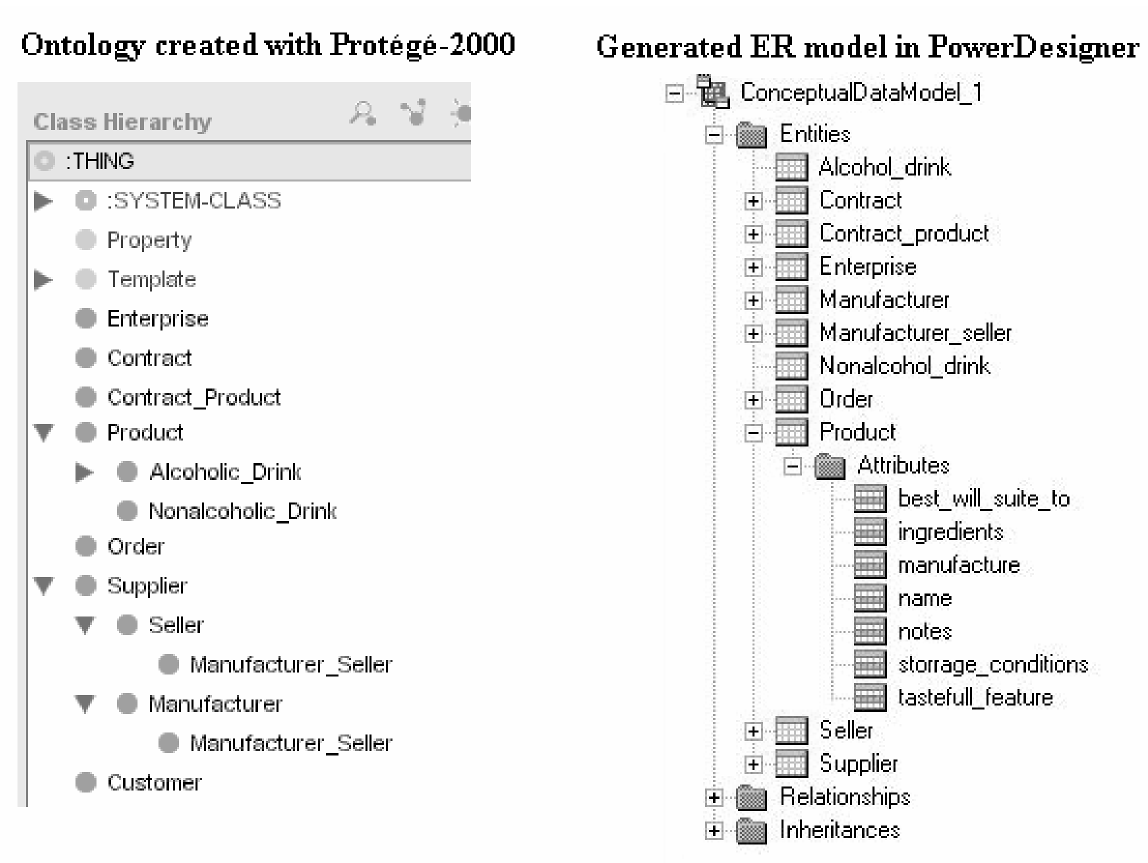

Figure 3: An example of automatic ontology transformation into a conceptual model.

CONCLUSIONS AND FUTURE WORK

The analysis of the related works on knowledge-based information systems development shows that enterprise knowledge represented by ontology can be transformed into conceptual model. This transformation is possible, since concepts of the meta-model of ontology can be mapped into concepts of the meta-model of conceptual model.

The approach for enterprise ontology transformation into conceptual model is proposed in the paper. We argue that the ontology can be automatically transformed into conceptual model, which can be transformed into physical model and consequently into a code, which can be executed on MS SQL, Oracle, DB2 and other modern DBMSs.

The prototype was developed and the experiment of ontology automatic transformation into conceptual model was carried out. Provided experiment shows that the suggested approach can be used for transformation of enterprise ontology into conceptual model.

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

REFERENCES

[1] Wand, Y., V.C. Storey, R. Weber. An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems (TODS), December 1999, vol. 24(4), pp. 494–528.

[2] Palol, X. de. A basic repository of operations for the refinement of general ontologies. Reseach Report LSI-04-26-R, UUPC, 2004.

[3] Guizzardi, G., H. Herre, G. Wagner. On the General Ontological Foundations of Conceptual Modeling. In Proc. of 21th International Conference on Conceptual Modeling (ER 2002). Springer-Verlag, Berlin, LNCS.

[4] Yang, X. Ontologies and How to Build Them. 2001. (March, 2006): http://www.ics.uci.edu/~xwy/publications/area-exam.ps.

[5] Gruber, T. What is an Ontology? (September, 2005): http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

[6] Bugaite, D., O. Vasilecas. Ontology-Based Elicitation of Business Rules (to appear). In A. G. Nilsson, R. Gustas, W. Wojtkowski, W. G. Wojtkowski, S. Wrycza, J. Zupancic (eds.): Information Systems Development: Proc. of the ISD'2004. Springer-Verlag, Sweden, 2006, pp. 795-806.

[7] Jarrar, M., J. Demey, R. Meersman. On Using Conceptual Data Modeling for Ontology Engineering. In S. Spaccapietra, S. March, K. Aberer (eds.): Journal on Data Semantics. 2003, LNCS (vol. 2800), Springer, pp. 185-207.

[8] Levendovszky, T., G. Karsai, M. Maroti, A. Ledeczi, H. Charaf. Model Reuse with Metamodel-Based Transformations. In C. Gacek (ed.): Proc. of ICSR-7. 2002, LNCS (vol. 2319/2002), Springer Berlin / Heidelberg, pp. 166-178.

[9] Ontology Definition Metamodel. OMG. 2005. (February, 2006): http://www.omg.org/docs/ad/05-08-01.pdf.

[10] Maidana, L. C. y C., R. G. y O. Chiotti. A Semantics Definition Metamodel. In M. Solar, D. Fernández-Baca, E. Cuadros-Vargas (rds.): Proc. of 30ma Conferencia Latinoamericana de Informática (CLEI2004). 2004, Sociedad Peruana de Computación, pp. 150-159.

[11] Halpin, T. An ORM Metamodel. Journal of Conceptual Modeling. Issue 16. October, 2000. (March, 2006): http://inconcept.com/JCM/October2000/halpin.html.

[12] Guarino, N. Formal Ontology and Information Systems. In Proc. of FOIS'98, Trento, Italy, 6-8 June, 1998. Amsterdam, IOS Press, pp. 3-15.

[13] Knublauch, H. UMLBackend. (April, 2006): http://protege.stanford.edu/plugins/uml/images/Protege-MetaModel.pdf

[14] Jakkilinki, R., N. Sharda, M. Georgievski. Developing an Ontology for Teaching Multimedia Design and Planning. (February, 2005): http://sci.vu.edu.au/~nalin/MUDPYOntologyPreprintV2.pdf.

[15] Grosso, W. The Protégé Axiom Language and Toolset ("PAL"). 2002. (September, 2005): http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.html.

[16] Bugaite, D., O. Vasilecas. Framework on application domain ontology transformation into set of business rules. In B. Rachev, A. Smirkarov (eds.): Proc. of the International Conference on Computer Systems and Technologies "CompSysTech'05", Varna, Bulgaria, 16-17 June, 2005, p. IIIB.8-1-IIIB.8-6.

ABOUT THE AUTHOR

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.

Justas Trinkunas, Department of Information Systems, Vilnius Gediminas Technical University, E-mail: justas@isl.vtu.lt.