

Application domain ontology-based transformation to database design*

Justas TRINKŪNAS, Olegas VASILECAS (VGTU)

e-mail: justas@isl.vgtu.lt, olegas@fm.vgtu.lt

Abstract. The paper analyses graph oriented ontology transformation into conceptual data model. A number of methods were proposed to develop conceptual data models, but only few deals with knowledge reuse. In this paper we present an approach for knowledge represented by ontology automatic transformation into conceptual data model. The graph transformation language is presented and adapted for formal transformation of ontology into conceptual model. Details and examples of proposed ontology transformation into conceptual data model are presented.

Keywords: ontology, conceptual modelling, transformation.

1. Introduction

Conceptual data modelling methodologies became well known and quite successful because of their methodological guidance in building conceptual models of information systems. Moreover, they provide a graphical representation of a model.

Conceptual data schemes, also called semantic data models, were developed to capture the meaning of an application domain as perceived by its developers [1, 2].

But there are the following main problems concerning conceptual modelling. Firstly, as discussed in [2], meanings of conceptual modelling constructs have to be defined rigorously to employ them effectively. Often, however, rigorous definitions of these constructs are missing. Secondary, in general, most conceptual schemas are developed from scratch, which means wasting previous efforts and time [3]. Thirdly, domain knowledge acquired in the analysis of some particular domain 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.

Companies install information systems to increase the effectiveness of activities, to get more profit and increase the value of the company. However to develop database today is hard and much time required work. When database designer creates new database he has to solve the same analytical problems every time.

*The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project “Business Rules Solutions for Information Systems Development (VeTIS)” Reg. No. B-07042.

Every application domain has a lot of terms and business rules which database designer has to understand. More over, a database designer may not have knowledge of the domain in that is being modelled then thus must rely on the user to articulate the design requirements. Also, one term can have many titles and meanings. This could be very aggravating circumstances for getting requirements for database. Database designer has to anticipate what will be the cycle of existence of the system, how it can change in the future, what are the threats and weaknesses of the system and how to avoid them. And finally, the fast changing requirements are the main problem of creating and/or modifying applications. Most of these requirements are related to business rules. In a case of change of application domain, database designer has to adopt the database quickly and effectively.

To solve all these problems and to facilitate the job of database designer databases, which are based on knowledge bases and its' management systems, was began to use. Ontologies were begun to use for classification and formalization of business rules. Ontology – particular kind of knowledge base that describes the facts which are considered as always right by some group of users. Ontologies are used to capture knowledge about some domain of interest. It describes the concepts in the domain and also the relationships that hold between those concepts.

This kind of database design method is not used frequently, because it is in early stage of evolution. This work shows how domain ontology can be used for eliciting of business rules and speed up the design stage of information systems. It describes how the work of database designer could be facilitated and suggests how the process of ontology transformation to conceptual model could be improved or upgraded.

A database design stage is essential in every IT company's activities. The problem is that the process of database design is complicated and much time required job. Researches around the world are trying to find new ways of making that process more comfortable for designer. Including the knowledge base into database modelling stage could be helpful for designers. It could speed up the design stage and help in modifying databases.

In this work a knowledge base in a form of ontology has to be made. The main purpose of this work is to upgrade the process of application domain ontology transformation to conceptual model. The goal will be reached by making smaller tasks:

2. Related work

According to the authors of the book "Handbook on Ontologies" [4], ontologies are becoming of increasing importance in fields such as knowledge management, information integration, cooperative information systems, information retrieval and electronic commerce. One application area which has recently seen an explosion of interest is the so called Semantic web, where ontologies are set to play a key role in establishing a common terminology between agents, thus ensuring that different agents have a shared understanding of terms used in semantic mark-up.

Ontology development is fundamentally a difficult problem. Research on creating ontologies and using ontologies has been motivated by the Semantic Web and knowledge reuse. The Semantic Web is intended to extend the World Wide Web by capturing

and representing the semantics of an application domain in domain ontologies. Research on the Semantic Web investigated the need for libraries of ontologies, the most well-known of which is the DAML ontologies that were developed specifically for the Semantic Web.

Another motivation for the development of ontology libraries is for knowledge reuse, which is important because of the overwhelming amount of information that is constantly being generated. It is intended that such ontology libraries will be publicly available [5].

In the source [6] is stated, that strategy for building ontologies would be to reuse large ontologies like SENSUS (with more than 70,000 concepts) to build domain specific ontologies and knowledge bases. The same ontology can be used for building several knowledge bases, which would share the same skeleton. Extensions of the skeleton should be possible at the low level by adding domain-specific sub concepts or at the high level by adding intermediate or upper level concepts that cover new areas. If systems are built with the same ontology, they share a common underlying structure, therefore, merging and sharing their knowledge bases and inference mechanisms will become easier.

In this paper it is analyzed the development of actual ontologies. Most of ontologies are developed anew without using available knowledge. To solve this problem the usage of relation data bases was chosen. Ontologies, their types, development methods, tools, languages are presented in this paper. According to the analysis the rules of transformation from concept data model into ontology are suggested. The process of performed experiment, results and findings are presented in this paper.

This is the main question which has to be answered before going into ontology design stage. In the document called "A Guide to Creating Your First Ontology" [7] about the creation of ontology authors discuss some of the reason and explanations why do we need ontology at all. Some of these reasons are:

- to share common understanding of the structure of information among people or software agents;
- to enable reuse of domain knowledge;
- to make domain assumptions explicit;
- to separate domain knowledge from the operational knowledge;
- to analyze domain knowledge.

3. Proposed approach

Mathematically we define ontology using graph formalism. In work [8] authors define an ontology O as a directed labelled graph $G_O = (N, E)$ where N is a finite set of labelled nodes and E is a finite set of labelled edges. An edge e is written as a triplet $(n1, \alpha, n2)$ where $n1$ and $n2$ are members of N and α is the label of the edge. The structure of graph consisting from [9]:

- a set of concepts (vertices in a graph);
- a set of relationships connecting concepts (directed edges in a graph);
- a set of instances assigned to a particular concepts (data records assigned to concepts or relation).

ER model can be represented as a graph. We define ER model using graph formalism. ER model is a directed labelled graph $G_{ER} = (N, E)$ where N is a finite set of labelled nodes and E is a finite set of labelled edges. An edge e is written as a triplet $(n1, \alpha, n2)$ where $n1$ and $n2$ are members of N and α is the label of the edge.

We have adopted graph transformation language used in [8] work. The language consists of the five basic operations: node addition, edge addition, node deletion, edge deletion, and abstraction. Currently we need only two operations (node addition and edge addition).

Node Addition. Given the graph G , a node N and its adjacent edges $\{(N, \alpha i, m j)\}$ to add, the node addition results in a graph $G' = (M', E')$ where $M' = M \cup N$ and $E' = E \cup \{(N, \alpha i, m j)\}$.

Edge Addition. Given a graph G and a set of edges $SE = \{(m i, \alpha j, m k)\}$ to add the edge addition operation $EA[G, SE]$ results in a graph $G' = (M, E')$ where $E' = E \cup SE$.

The node addition operation can be used to introduce new objects into ontology from the conceptual data model. The edge addition operation is needed to build relationships between ontology elements.

Now we describe how ontology can be transformed to conceptual data model using graph formalism based on metamodelling.

Transformation from ontology G_O to conceptual data model G_{ER} can be presented as:

$$G_O \rightarrow G_{ER}, \quad (1)$$

where G_O is an ontology represented as graph which is based on OWL metamodel,

G_{ER} is ER model represented as graph based on ER metamodel,

\rightarrow is transformation of the elements of graph which consists of Node Addition and Edge Addition operations defined above. All elements of ER are transformed into OWL ontology elements.

4. Case study

After deep analysis of types of contracts in Lithuania the following diagram was created. The Fig. 1 represents logical structure of types of contracts in Lithuania.

Briefly we describe proposed method of building conceptual model from the OWL DL ontology. The method consists of four main steps:

1. The first step is knowledge acquisition from the word, documents, people, conceptual data models, ontologies and other sources. All extracted knowledge is written in the domain ontology in OWL DL format. We use Protégé 3.3 tool, however other tool could be chosen for ontology development. Domain ontology is created manually. But we are expanding our work and in near future we will propose semiautomatic method for ontology development from existing conceptual models, ontologies and other sources.
2. The second step is the transformation of domain ontology into conceptual data model with our plug-in OntER. Created conceptual data model can be opened with Sybase Power Designer 12.0 tool and adapted for your needs.

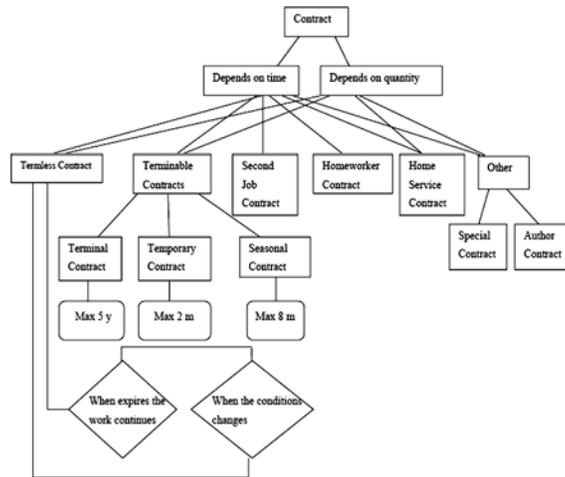


Fig. 1. Logical structure of types of contracts in Lithuania [15].

3. The third step is verification of conceptual data model. If we made changes with Power Designer 12.0 we need to verify if conceptual data model is valid. The conceptual data model is compared with the domain ontology.
4. The last step is the generation of physical data model with Power Designer 12.0 for a particular DBMS. This feature is already implemented in the original version of Power Designer 12.0.

Through a simple generation procedure, you can transfer the solid design framework of the conceptual data model to the physical data model. The physical data model adapts your design to the specifics of a DBMS and puts you well on the way to complete physical implementation.

5. Conclusions and future works

We presented graph oriented model for ontology transformation into data conceptual data model based on metamodels. After deep analysis of types of contracts in Lithuania the Payroll ontology was created using Protégé tool. The experiment showed that creating conceptual data models from ontology brings great benefits. The main advantage of proposed method is formally defined transformation of ontology to conceptual data model. However, it is not possible to transform all elements from OWL DL ontology into conceptual data model straight forward because OWL DL is semantically richer.

References

1. R. Meersman, The use of lexicons and other computer-linguistic tools, in: *Proc. of the International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 99)*, Y. Zhang *et al.* (Eds.), Heidelberg, Springer Verlag (1999), pp. 1–14.
2. Y. Wand, V.C. Storey, R. Weber, An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems (TODS)*, **24**(4), 494–528 (1999).

3. X. Palol, A basic repository of operations for the refinement of general ontologies, *Reseach Report LSI-04-26-R*, UUPC (2004).
4. S. Steffen (Edt), *Handbook on Ontologies*, Springer (2004).
5. V. Sugumaran, V.C. Storey, The role of domain ontologies in database design: an ontology management and conceptual modeling environment, *Citeulike*, **31**(3), 1064–1094 (2006).
6. O. Corcho, M. Fernandez–Lopez, A. Gomez–Perez, Methodologies, tools and languages for building ontologies. Where is their meeting point?, *Data & Knowledge Engineering*, **46**, 41–64 (2003).
7. N.F. Noy, D.L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford University, Stanford, CA, 94305 (2000).
8. P. Mitra, G. Wiederhold, M. Kersten, A graph oriented model for articulation of ontology interdependencies, *Proc. Extending DataBase Technologies*, Springer, Berlin Heidelberg, LNCS 1777, 86–100 (2000).
9. J. Davies, R. Studer, P. Warren, *Semantic Web Technologies Trends and Research in Ontology-based Systems*, John Wiley & Sons Ltd (2006).

REZIUMĒ

J. Trinkūnas, O. Vasilecas. Dalykinės srities ontologijos transformacija į koncepcinį modelį

Straipsnyje analizuojama ontologijos transformacija į koncepcinį duomenų modelį. Atlikus literatūros analizę, nustatyta, kad egzistuoja daugybė metodų, kaip kurti koncepcinius modelius, tačiau tik keli iš jų siūlo pakartotinai panaudoti turimas žinias. Šiame straipsnyje pristatome koncepcinio modelio kūrimą panaudojant ontologiją.