Eder J. & Koncilia C.(2005). *Interoperability in Temporal Ontologies*. Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP'05 - Co-located with CAiSE'05 Conference), June 13-14, Porto, Portugal, CEUR-WS.org.

Interoperability in Temporal Ontologies

Johann Eder, Christian Koncilia

University of Klagenfurt
Dep. of Informatics-Systems
{eder, koncilia}@isys.uni-klu.ac.at
March 2005

Abstract. In this position paper, we will discuss the interoperability between different versions of a system (an ontology). In particular, we briefly present and analyze our approach to deal with changes in ontologies. Furthermore, we discuss two open questions in the area of temporal ontologies: a) The definition of Inter-Structure Ontologies to describe the changes between two versions of an ontology and b) the need to take semantics into account in a temporal ontology.

1. Introduction

In general interoperability is all about different (software) systems working together. One aspect of interoperability is, therefore, the ability to exchange different kinds of data between different kinds of software systems. In this paper, we will discuss another type of interoperability: the ability of different versions of a system (an ontology) to work together.

Ontologies are seen as a promising approach for adding semantics to data processing. An ontology is defined as a shared conceptualization of a certain domain [3]. It describes the concepts of a domain, their properties and their relationships. Much work has already been done to analyse multiple heterogeneous ontologies, their integration and their coexistence.

Surprisingly little attention was drawn to the fact that the reality an ontology describes and/or the view of the observers sharing the conceptualization on the reality may change.

There are three different basic approaches of how to deal with such changing knowledge. First of all, we could simply ignore modifications, and describe the world in a completely static way. Obviously, this approach is of limited suitability for real world applications. The second approach is a little bit more sophisticated: by adopting our knowledge description we always represent and store the most recent version of knowledge. This is the most frequent approach nowadays. It has the disadvantage that we lose knowledge about the past. The third approach takes into account that the knowledge about modifications is again knowledge that may be important. In this approach we would have to describe different versions of knowledge and the relations between these versions. The last two approaches are well known in the temporal database community. The first one is called (Schema) Evolution, the latter one (Schema) Versioning [4].

In [2] we presented how a simple ontology description formalism, namely a directed graph, has to be extended to represent changing knowledge, and which extensions to such a "specification language" would be meaningful.

In this position paper we briefly discuss our approach for a temporal ontology and two open questions in the area of temporal ontologies: a) The definition of Inter-Structure Ontologies to describe the changes between two versions of an ontology and b) the need to take semantics into account in a temporal ontology.

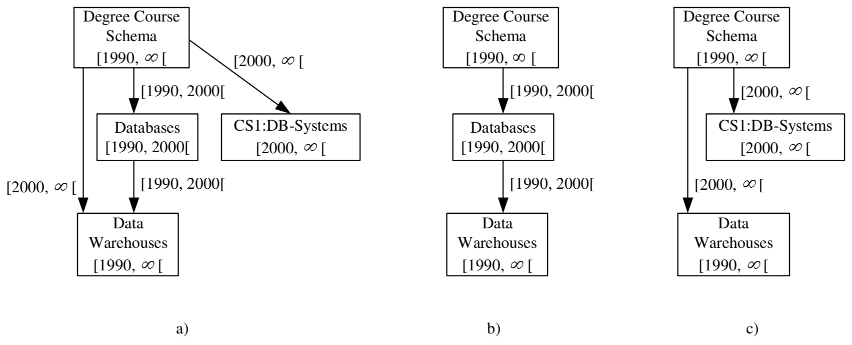
Degree Course Schema [1990, ∞ [
[2000, ∞ [
[1990, 2000[
Databases [1990, 2000[
CS1:DB-Systems [2000, ∞ [
[2000, ∞ [
[1990, 2000[
Data Warehouses [1990, ∞ [
a)
Degree Course Schema [1990, ∞ [
[1990, 2000[
Databases [1990, 2000[
[1990, 2000[
Data Warehouses [1990, ∞ [
b)
Degree Course Schema [1990, ∞ [
[2000, ∞ [
CS1:DB-Systems [2000, ∞ [
[2000, ∞ [
Data Warehouses [1990, ∞ [
c)

Fig. 1. a) An ontology versioning graph and its deduced versions b) and c)

2. Our Approach

Nowadays, several languages exist to specify an „explicit specification of a conceptualization of a domain", e.g., DAML+OIL, OWL or CL. Another possibility to specify such a conceptualization is to use a graph where nodes represent concepts, and edges represent the relations between two concepts [5].

In order to support a temporal extension, our model uses a linear and discrete model of time. Furthermore, it supports time stamps for all nodes and edges in the graph. These time stamps represent the Valid Time of the corresponding element. Valid time defines the time, in which a fact (in our model a fact may be both, a node or an edge) is true in the real world [4]. A fact may have more than one time points or time intervals during which it is true in the modelled world. These time stamps are defined as [Ts, Te[where Ts is the beginning of the valid time, Te is the end of the valid time. We represent that a fact is valid until now by Te = 4. Please note that we use the syntax [A, B[to represent a half-closed interval. In this half-closed interval the instant A is included, whereas the instant B is excluded.

Our model enables us to represent different versions of an ontology in a graph. We called this graph ontology versioning graph. In [2] we gave a formal definition of such an ontology versioning graph. In this paper we confine ourselves to describe this model intuitively:

An ontology versioning graph consists of a set of class definitions (nodes) and a set of binary relation definitions (edges). Each class definition has a label, a set of slots (attributes and properties assigned to a class) and a valid time [Ts, Te[. Furthermore, each relation definition has an assigned valid time [Ts, Te[.

Furthermore, we formally defined some versioning operations in [2] on this graph model to INSERT, UPDATE and DELETE classes and relations. INSERT inserts a new class / relation into the graph. UPDATE sets the end of the valid time of the corresponding class / relation to a new value, and inserts a new version of this class / relation. DELETE sets the end of the valid time of the most recent version (the version where Te = 4) of the corresponding class / relation to a new value.

2.1. Selecting a Specific Ontology Version

The ontology versioning graph defined in our approach consists of all possible ontology versions. Or, in other words, we do not define several ontologies where each ontology represents a version of an ontology. In fact, we define a single ontology which is composed of all ontology versions

Figure 1 a) shows an example of an ontology versioning graph. As can be seen, this ontology versioning graph consists of two versions b) and c). In this example, version b) is valid during the interval [1990, 2000[and version c) with a valid time [2000, 4[, where [1990, 2000[represents that this version is valid from 1990 until 1999 (2000 is not included as we use half-closed intervals).

Intuitively we can say that if we represent all timestamps [Ts, Te[of all temporal components within our ontology versioning graph on a linear time axis, the interval between two consecutive timestamps on this axis represents the valid time of an ontology version.

In [2] we formally defined how to select a specific ontology version by choosing a particular time point. We also discussed the concept of stable intervals. Intuitively, we can say that such a stable interval is a view defined on an ontology versioning graph that is valid for a given time interval [Ts, Te[. All classes and relations within this ontology versioning graph are also valid for the given time interval. In other words, within such a stable interval there cannot exist different versions of classes or relations. In the example shown in Fig. 1 we have two stable intervals: the first is valid during the interval [1990, 2000[, the second one during the interval [2000, 4[.

3. Further Work

There is still a lot of further work that has to be done in the area of temporal ontologies. We will now briefly discuss two open questions that we currently work on:

3.1. Temporal Integrity Constraints

Consider for example a *Part-Of* relation between two concepts, e.g., a *Table* and the *Surface* of this table. The type of this relation, i.e. Part-Of, has a wide influence on the temporal integrity constraints. In this example, you cannot remove the surface from table without destroying the table. In other words, if you remove it the table is no longer a table, and the surface no longer a surface. Hence, the valid time of the surface has a direct influence on the valid time of the table. Including temporal integrity constraints into temporal ontologies naturally extends to change propagation and truth maintenance in otology versions.

3.2. Ontologies of Change

Another important topic is the description of changes between different ontology versions. As we described before, an ontology versioning graph consists of several different versions of an ontology. The question is how to describe changes between two versions of an ontology. For instance, in Fig. 1 we could describe that the *Databases* lecture (see Fig. 1 a)) has been renamed and is now called *CS1:DB-Systems* (see Fig. 1 b)).

Our idea is to use an ontology to describe changes between two versions of an ontology. Such a description would again lead to some interesting questions, for instance: If we have an ontology that describes the changes between version A and version B of an ontology, and an ontology that describes changes between version B and version C of the same ontology, can be deduce knowledge about what changed from version A to version C?

4. Conclusions

Temporal ontologies are a concept for managing the change of admitted terms and the change of meaning. With ontology versioning graphs results from schema evolution and versioning can be adopted for dealing with evolving ontologies. Application possibilities are numerous: from the annotation of changed meanings when reading old documents to automatic transformation of data.

References

[1] O. Etzion, S. Jajodia, and S. Sripada, editors. Temporal Databases: Research and Practise. Springer-Verlag (LNCS 1399), 1998.

[2] J. Eder and C. Koncilia. Modelling Changes in Ontologies. Springer-Verlag (LNCS 3292), 2004.

[3] T. Gruber. A Translation Approach to Portable Ontology Specification. In Knowledge Acquisition 5(2):199-220. World Wide Web Consortium (W3C), 2003.

[4] C. S. Jensen and C. E. Dyreson, editors. A consensus Glossary of Temporal Database Concepts - Feb. 1998 Version, pages 367–405. Springer-Verlag, 1998. In [EJS98].

[5] P. Mitra, G. Wiederhold, and M. Kersten. A Graph-Oriented Model for Articulation of Ontology Interdependencies. In Proceedings Conference on Extending Database Technology 2000 (EDBT'2000), Konstanz, Germany, 2000, volume LNCS: 1777, pages 86+, 2000.