Bridging the Semantic Gap between Ontology Versions

Tomi Kauppinen and Eero Hyvönen

University of Helsinki, Department of Computer Science
P.O. Box 68 (Gustav Hällströminkatu 2b)
FI-00014 University of Helsinki, Finland
firstname.lastname@cs.helsinki.fi
http://www.cs.helsinki.fi/group/seco/

Abstract. Ontologies evolve over time, when they are altered to correct errors, to accommodate new information or to adjust the representation of the domain. Hence there is a need for methods and means to manage the ontology evolution in order to ensure that applications using different versions of an ontology remain compatible with respect each other and metadata annotated. In this paper an ontology versioning framework is presented to enable ontology versioning. The presented approach aims to provide means for expressing the relations between modified concepts after a change has occurred, and in this way to enable compatibility and reasoning based on an ontology version history. We propose change bridges to be used as mappings between resources in successive ontology versions.

1 Introduction

1.1 Background

Ontologies provide means for explicating concepts of the world and relationships between them. This is done by conceptualizing knowledge about the world into an abstract, simplified form that we wish to use for some purpose. Formally an ontology is an explicit specification of a conceptualization [3]. It thus specifies explicitly a representation of a piece of conceptualized knowledge. In order to do this, ontologies employ the notion of class hierarchy, where the subclasses of a superclass inherit properties defined for the superclass. In addition, subclasses may have additional properties.

The Semantic Web [1, 7] is based on using shared ontologies in a standard way, which requires the ability to use and to re-use ontologies in different applications at different times. However, ontologies evolve over time: they are altered to correct errors, to accommodate new information or to adjust the representation of the domain as the world changes [6]. Hence there is a strong need to revise ontologies. A method is needed for reconciling the different ontology versions with each other, since different versions of the ontology are used at different times. Otherwise differences in the ontology versions will disable the ability of an application to work properly with other systems and metadata based on another ontology version.

1.2 A Motivating Example

For example, assume the location ontology of Figure 1 that describes some places of the world during two successive periods of time. The nodes in the RDF-graph describe countries and places as geographical overlapping areas, and the edges represent the *part of* relation. For example, *Finland* is a part of *Europe* and *Petsamo* is a part of *Finland* in the version Ov_1 on the left. In the new version Ov_2 on the right, *Petsamo* has the new name *Pechenga* and it is a part of *Russia* as a result of the World War II (and the fall of the Soviet Union later). In Finnish museum collections, there are lots of items annotated using *Petsamo*, say a Lappish knife used in the area during the 1920's. If the current ontology Ov_2 is used for retrieving information annotated using the historical notion of *Petsamo* in Ov_1, data would be lost without knowing about the change in the ontology.

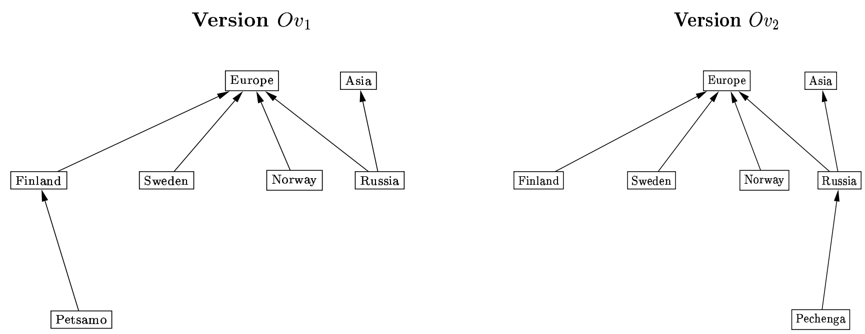

Fig. 1. Ontology versions Ov_1 (before the World War II) and Ov_2 (after the World War II). The directed edges represent part of -relations between the concepts.

In this work we present a framework [8] that aims to provide means for reasoning based on a complete versioning history. The framework includes the generic notion of *change bridge* for describing ontology resource chances, and a basic set of particular change bridge types that constitute the class hierarchy of a *change bridge ontology*. Ontology changes are represented as instances of the change types relating concepts in successive ontology versions with each other. The change bridging ontology is represented using the Resource Description Framework (RDF) [2].

2 Change Bridges Defined

Let us assume two versions Ov_1 and Ov_2 of a location ontology defining the concepts *Germany*, *East Germany* and *West Germany* during different time periods as depicted in Figure 2. The new version Ov_2 of the ontology is not backward

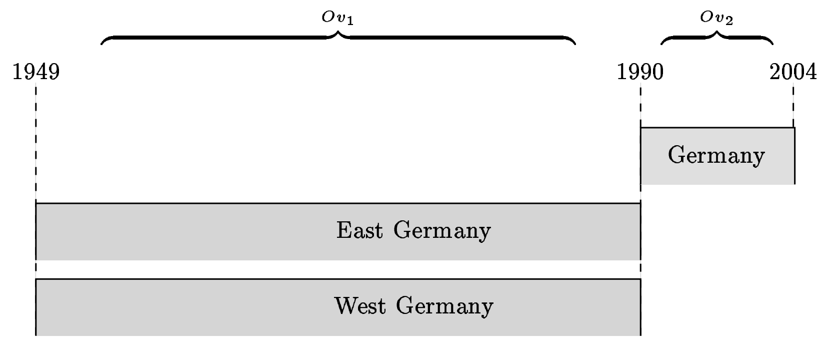

Fig. 2. Classes *West Germany*, *East Germany* and *Germany* in ontology versions Ov_1 and Ov_2.

compatible [5] with Ov_1 since *Germany* is not present in 1949-1990, and Ov_1 is not forward compatible with Ov_2 since *East Germany* and *West Germany* disappear after 1990. However, it would be essential to use multiple versions of the ontology in reasoning about the resources. In our example, the user might be interested in German wines that were classified before under *East Germany* or *West Germany* and are nowadays annotated with *Germany*. The requirement for backward compatibility in a strict logical sense [5] cannot be met easily in practical world ontologies like this, where there are a lot of different revising needs for an ontology, due to changes in the real world or due to corrections and modifications needed as time goes by.

Alternatively, less formal but more flexible ontology mapping descriptions can be used for reasoning about an evolved ontology. Mappings between ontologies have been discussed e.g. in [9, 11, 4]. The focus of research in ontology mapping has been to investigate how different ontologies can be aligned with each other for interoperability. Our goal is related but different: we focus on aligning the revisions of a single ontology in time. The revision mappings have to be identified and used to create bridges between those resources of ontology revisions Ov_1 and Ov_2 that the change has touched. We call such a mapping a *change bridge*. A change bridge is a mapping between resource sets Rv_1 and Rv_2 of two successive ontology versions Ov_1 and Ov_1, respectively. It defines how Rv_1 relate with Rv_2. Change bridges are individuals of different *change bridge classes*. If there are many changes between Ov_1 and Ov_2, a set of change bridges called a *version bridge* can be used to express all of them. For example, in Figure 3 the semantic change between the versions Ov_1 and Ov_2 of Figure 1 is expressed as an instance of a change bridge called *usedToBe*.

In order to help identifying possible bridges and how they should be used, the following questions are to be answered:

- What has changed (in the old version Ov_1)?
- What has it changed into (in the new version Ov_2)?

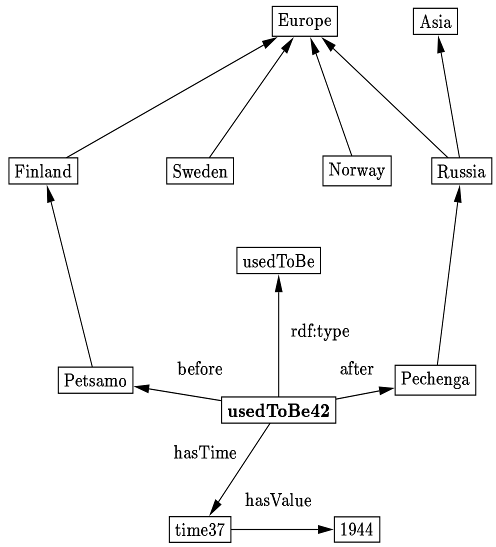

Fig. 3. Combination of ontology versions Ov_1 and Ov_2 where the relation between *Petsamo* and *Pechenga* is expressed using a change bridge. Notice that *usedToBe42* is an instance of the change bridge *usedToBe*.

- How can the change be explicitly expressed as of change bridge between Ov_1 and Ov_2?
- How can the bridge be used to reason about the related concepts?
- How can the whole ontology version history be used to reason about the related concepts?
- How can different ontology versions be determined based on bridges?

3 Expressing Change and Version Bridges

A revision bridge can be expressed as a set of individual bridges conforming to a change bridge ontology.

There are basically two choices how a change bridge could be expressed in an evolving ontology: within the location ontology versions themselves or as a separate annotation conforming to the bridge ontology. Separating the change descriptions from the evolving ontology seems more natural because ontology changes are conceptually higher level metadata about the evolving ontology. Figure 4 illustrates the situation. Node *usedToBe73* is an instance of the bridge change class usedToBe telling that *Myanmar* changed into *Burma* in 1989.

To take another example, Figure 5 depicts the merger of East and West Germany (Figure 2). The instance *merged42* of the merged bridge tells that *East Germany* and *West Germany* were merged into *Germany* in the new version in

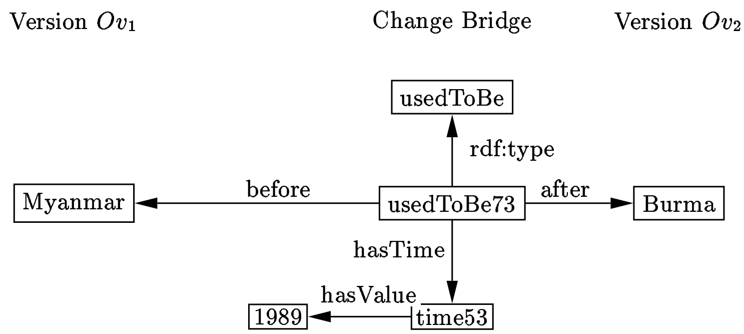

Fig. 4. Example of a *usedToBe*-bridge between two ontology versions Ov_1 and Ov_2.

1991. Again, separate metadata annotation is added, and the ontology versions remain untouched.

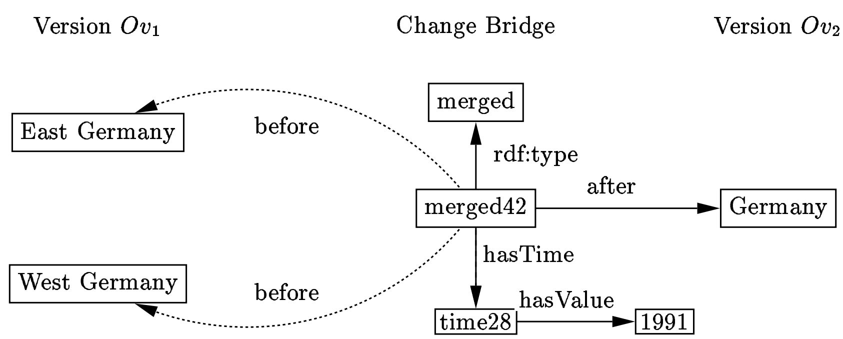

Fig. 5. Usage of *merged*-bridge from the change bridge ontology.

Figure 6 presents still another usage example of a bridging class, the *split*-bridge. Here the resource *Czechoslovakia* has been split into two distinct resources *Czech Republic* and *Slovak Republic* in the revised ontology. The *split*-bridge describes the newly formed relation between the resources related to the change.

Table 1 list the classes of the first version of our change bridge ontology.

resourceChange is a bridge intended to be used when something has happened between classes (or individuals) of versions Ov_1 and Ov_2 of the ontology. Typical bridges of this kind are *merged*, *split* and *sameAs*. To form these bridges we have to know which classes (or individuals) are present in versions

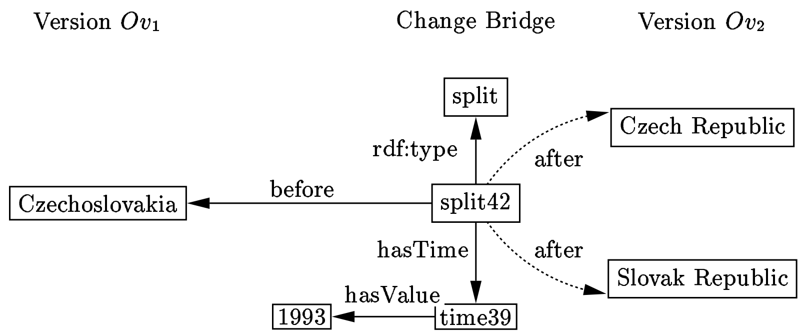

Fig. 6. Usage of split-bridge of the change bridge ontology.

Ov_1 and Ov_2 or merely the differences between classes (or individuals) in Ov_1 and those in Ov_2.

hierarchyChange combines different bridges defining the manipulation of the class hierarchy. In other words, in this case versions Ov_1 and Ov_2 of the ontology have a difference in their hierarchies but they can still have exactly the same classes in them. Typically, hierarchy alteration occurs when classes or properties are moved down or up in the hierarchy or subclass-superclass relations are modified.

propertyChange describes the situation where some properties of classes of Ov_1 have been altered to form a new version Ov_2 of the ontology. A typical example of a property change is the *samePropertyAs*-relation defining the similarity of two distinct properties.

typeChange is a crucial bridge, defining a mapping between those classes and instances of Ov_1 and classes and instances of Ov_2 where a re-classification (class to instance or vice versa) has occurred.

The classes in the change bridge ontology are an adaptation of the different change operation types presented by Noy and Klein [10]. However, in contrast to the change operation types, classes in the change bridge ontology do not try to express changes such as deletion of a class, addition of a class, etc., but rather relations between revised resources in successive ontology versions. The bridge classes form a simple subclass-hierarchy depicting the different roles and types of bridges.

Simple usage rules for change bridge ontology are needed, describing how it can be used to form mappings between ontology versions. These rules include:

- Relations between resources in versions Ov_1 and Ov_2 are expressed using the change bridge ontology by creating instances of its classes.
- The bridges are stored in a separate annotation file.
- The arcs point from the bridge class instances to resources in versions Ov_1 and Ov_2 of the ontology.

Change type	
resourceChange	resourcesDeclaredDisjoint differentFrom merged sameAs split usedToBe
hierarchyChange	classMovedDown classMovedUp propertyMovedDown propertyMovedUp subclassSuperclassLinkAdded subclassSuperclassLinkRemoved
propertyChange	narrowedPropertyRestriction samePropertyAs widenedPropertyRestriction
typeChange	classRe-classifiedAsInstance instanceRe-classifiedAsClass setOfPropertiesEncapsulatedIntoNewClass

Table 1. Concepts of the change bridge ontology.

- Mappings can be made either between the entities of versions Ov_1 and Ov_2 of the ontology or between the entities found only in Ov_2. In other words, the ontology modeler can leave — if he chooses so — the outdated classes also to the new version and provide bridges with the more recent classes.
- When a mapping is made, it has to be complete, that is, no halfway bridges having only partial information are allowed.
- A revision ontology is used to automatically get identification, status, author and other important identification information from the versioning system in use.

3.1 Representing Partial Overlap

In the above examples, it was assumed that the bridges map the territory areas between ontology versions precisely. For example, East and West Germany were merged to form Germany exactly. There are situations, however, where a resource in the previous ontology maps only partly on the resources of the revised ontology. Consider, for example, that resources *Soviet Union* and *Russia* are used in a location ontology. If we want to express the fact that Soviet Union mostly changed into Russia, without extending our ontology to include all Soviet states that became independent at the same time, then the problem is that these resources are partly different and cannot be bridged with a sameAs bridge. For situations of this kind the definition of the change bridge could be extended with addition metadata expressing the degree of overlap between the mapped resources.

Figure 7 depicts how this situation could be handled using a *hasCovering* property. If we are considering territorial overlap, the value of *hasCovering* can be calculated by comparing the overlapping territories that modern day Russia and the former Soviet Union share:

$$\frac{Territory(Russia)}{Territory(SovietUnion)} = \frac{17{,}075{,}200 sq.km}{22{,}274{,}900 sq.km} \approx 0.77$$

Other possible dimensions for "covers" overlap could be, for example, the number of inhabitants, length of land boundaries or the coastline, depending on the selected viewpoint.

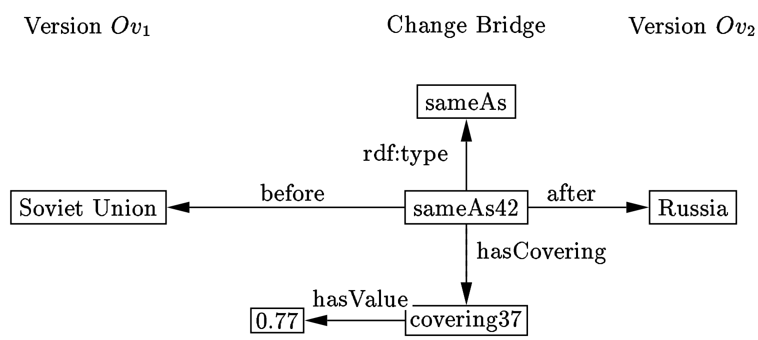

Fig. 7. The territorial overlap of Russia and the Soviet Union represented by the *hasCovering* property.

4 Conclusions

Ontologies are altered to correct errors, to accommodate new information or to adjust the representation of the domain. Hence there is a need for methods and means to represent ontology change and manage the ontology evolution in order to ensure that applications using different versions of ontologies and annotated data work properly. In this paper a framework for bridging the semantic gap between ontology versions was outlined.

The framework aims to issue several important issues in ontology engineering. First of all, it is important to explicate changes, in classes, instances, and their properties in an ontology. It is also necessary to identify what change operations have produced the changes and further express the change as a mapping between evolved entities. The idea of a change bridge was presented to enable bridging the semantic gap between different versions of an ontology.

We plan to develop the idea of change bridges further, and to apply it to building a Finnish temporal region ontology (Suomen Ajallinen PaikkaOntologia, SAPO). SAPO is planned to define different Finnish regions from the beginning of the 20th century and the various changes there have been over the time. Our goal is to exploit spatio-temporal change bridge knowledge in information retrieval applications. Future work also includes researching methods to automate the process of identifying possible bridges between ontology versions.

5 Acknowledgments

Discussions with Markus Holi are acknowledged. Our research was funded mainly by the National Technology Agency Tekes.

References

1. Tim Berners-Lee, Jim Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
2. D. Brickley and R. V. Guha. Resource Description Framework (RDF) Schema Specification 1.0. Candidate recommendation, World Wide Web Consortium, March 2000.
3. T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5, 1993:199–220, 1993.
4. A. Hameed, A. Preece, and D. Sleeman. Ontology reconciliations. In S. Staab and R. Studer, editors, *Handbook on ontologies*. Springer-Verlag: Heidelberg, Germany, 2004.
5. J. Heflin. Towards the semantic web: Knowledge representation in a dynamic, distributed environment. PhD Thesis, University of Maryland, 2001.
6. Jeff Heflin and James Hendler. Dynamic ontologies on the web. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 443–449. AAAI/MIT Press, Menlo Park, CA, 2000.
7. James Hendler, Dieter Fensel, Henry Liebermann, and Wolfgang Wahlster. *Spinning the Semantic Web.* MIT Press, 2002.
8. Tomi Kauppinen. An ontology versioning framework. Master's thesis, University of Helsinki, 2004.
9. Prasenjit Mitra, Gio Wiederhold, and Martin Kersten. A graph-oriented model for articulation of ontology interdependencies. *Extending DataBase Technologies, EDBT 2000, Konstanz, Germany, 2000*, 2000.
10. N. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems 5, 2003*, 2003.
11. N. F. Noy. Tools for mapping and merging ontologies. In S. Staab and R. Studer, editors, *Handbook on ontologies*. Springer-Verlag: Heidelberg, Germany, 2004.