Technological University Dublin

ARROW@TU Dublin

Conference papers

School of Computer Sciences

2005

Bi-directional Ontology Versioning BOV

Siyang Zhao
Technological University Dublin

Brendan Tierney
Technological University Dublin, brendan.tierney@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/scschcomcon

Part of the Computer Sciences Commons

Recommended Citation

Bi-directional Ontology Versioning BOV

Siyang Zhao and Brendan Tierney

School of Computing, Dublin Institute of Technology,
Kevin Street, Dublin 2, Republic of Ireland
Siyang.Zhao@comp.dit.ie, Brendan.Tierney@dit.ie

Abstract. This paper defines a new type of ontology versioning: Bi-directional Ontology Versioning: BOV. BOV provides bi-directional mappings and transformations between concepts in two ontology versions. BOV is identified by two levels mapping processes: linguistic mapping and structural mapping. BOV can satisfy the requirement of mapping in distributed environment.

Keywords: BOV, Bi-directional.

1   Introduction

Ontologies are increasing in popularity and show their importance in many fields such as knowledge engineering [1], knowledge representation [2], information integration [3], and etc. With the increasing uses of ontologies, a serious problem emerges: ontology changes and evolutions. Domain changes, adaptations of different tasks, or changes in the conceptualizations might cause modifications of the ontologies. Each change may create a new ontology version, which will cause many versions existing in the web. This may cause incompatibility between the data sources that use different ontology versions and give incorrect interpretations of data or make data inaccessible.

Therefore, ontology versioning is needed to handle the changes and the evolutions of ontologies. Ontology versioning is defined as "*the ability to handle changes in ontologies by creating and managing different variants of it*" [4]. This paper uses ontology versioning to describe the mappings between ontology versions.

Many data sharing and data reuse processes among distributed resources are bi-directional processes, which require bi-directional correspondences between ontology versions which describe the data of them. The existing ontology versioning approaches can only provide single-directional mappings between ontology versions, such as Ontoview [5] and PROMPTDiff [6]. When single-directional versioning approaches are used for bi-directional data sharing, they create two separated inverse single-directional mappings. This is time and effort consuming, because in these two separated inverse single-directional mapping processes, a lot of work is repeated, such as identifying mapping elements in different ontology versions. So a bi-directional ontology versioning is required for bi-directional data sharing and reuse.

Bi-directional Ontology Versioning (BOV) can relate the elements in different ontology versions in both directions at the same time and provide bi-directional transformation between them [7]. The major advantage of the bi-directional mapping is

that less time and effort are required for mapping identification and transformation in bi-directions.

2   Bi-directional Ontology Versioning (BOV)

The data sharing and data reuse processes are bi-directional processes. The single-directional ontology versioning and mapping are time-consumed and effort-consumed and can not fully satisfy the requirement of data sharing and data reuse. This paper has defined a new type of ontology versioning: Bi-directional Ontology Versioning BOV. BOV can improve the ability of data sharing and data reuse between different data resources that use different versions of the same ontology.

BOV contains four components: mapping elements, bi-directional mapping relation, bi-directional transformation and metadata. Figure 1 shows the structure of BOV.

Mapping Element			Mapping Element		
Ontology Version Title	Hierarchy	Label	Ontology Version Title	Hierarchy	Label
Bi-directional Mapping Relation					
Bi-directional Transformation					
Metadata					

Fig. 1. Bi-directional Ontology Versioning (BOV)

Definition 1. Bi-directional Ontology Versioning (BOV)
The bi-directional ontology versioning can be formalized by the 5-tuple $< E_1, E_2, R, T, M>$

- *E_1 and E_2: two mapping elements from different ontology versions. The mapping elements contain the label and the hierarchy of the element and ontology version's title.*
- *R: the bi-directional mapping relation between the mapping elements, whose value is an element of the set {Equivalence, Isomorphic, Isomerous, Multiple_change}.*
- *T: the bi-directional transformation, which transforms the mapping elements between each other in both directions.*
- *M: the metadata about mappings.*

2.1   Mapping Elements

The mapping elements indicate what elements from two ontology versions can be related and transformed. Figure 1 shows that the mapping element in BOV includes two elements from different ontology versions. These two elements have the same structure. The mapping element in BOV has three parts:

- Ontology version title: it describes the version title of the ontology version that the mapping element belongs to. It is used to identify the ontology version.
- Hierarchy: it includes all the ancestors of the mapping elements. Hierarchy is used to identify the mapping element in ontology versions.
- Label: label of the mapping element is used to describe the name of the mapping element and identify the mapping element.

2.2   Bi-directional Mapping Relation

Four bi-directional mapping relations are defined to describe the relationship between mapping elements: Equivalence, Isomorphic, Isomerous and Multiple_Change.

Definition 2. *Bi-directional mapping relations*
Suppose that there exist two ontology versions V1 and V2. Suppose also that E1 is an element in V and E2 is an element V2. L(element) expresses the label of an element. S(element, version) is the set that includes all the directly related elements and all the ancestors of the mapping element in ontology version.

- *Equivalence: When L(E1)=L(E2) and S(E1, V1)=S(E2, V2), the relationship between E1 and E2 is "Equivalence".*
- *Isomorphic: When L(E1)≠L(E2) and S(E1, V1)=H(E2, V2), the relation between E1 and E2 is "Isomorphic". The uses of the synonyms and different name representations would produce the "Isomorphic" relation between elements in two versions*
- *Isomerous: When L(E1)=L(E2) and S(E1, V1)≠S(E2, V2), the relation between E1 and E2 is "Isomerous". A "Isomerous" relation would involve addition or deletion of a concept, changes to the properties of a concept, etc.*
- *Multiple_change: When L(E1)≠L(E2) and S(E1, V1)≠S(E2, V2), the relation between E1 and E2 is "multiple_change". It means that there are linguistic changes and structure changes between E1 and E2.*

The set of bi-directional mapping relations is complete and non-overlapped. Complete here means they can describe all the relationship between the mapping elements. Non-overlapped here means the mapping cases they describe won't be overlapped, that is, one mapping case has and only has one bi-directional mapping relation for it.

2.3   Bi-directional Transformation

Transformation between elements is very important for the data sharing and reusing. It is only by transformation that ontology versions can understand and efficiently share the data of each other. The bi-directional transformation can help the data resources that use different ontology versions understand each other and freely share and exchange data.

The bi-directional transformation expression contains a bi-directional transformation expression. The transformation expression describes the transformation between the mapping elements. Usually the bi-directional transformation only supports 1:1 transformation if there is no transformation metadata supplied, because the system can not automatically extract two variants from one without any addition information.

2.4 Metadata

Metadata shows some additional information about the elements and change of ontologies such as data, author, and purpose of the change of versions. It is important for some data-sharing situations. For example, if there are two telephone companies want to share the data of the call tariff between each other, then the metadata to describe the call time is very important, because different time has different price of calls. The metadata would be supplied by the domain expert and the user, such as time and date, background and the task-dependent transformation metadata.

3 Mapping Identification

Two levels mappings are used to identify BOV: linguistic mapping and structural mapping. Linguistic mapping is used to compare the labels of elements and compute the linguistic similarity between elements. The result of the linguistic mapping is a linguistic similarity matrix. The algorithm is based on the *edit distance*, which is described in [8]. The *edit distance* measures the minimum number of token insertions, deletions, and substitutions required to transform one string into another using a dynamic programming algorithm. The Linguistic Similarity LS is computed as:

Suppose there are two sets of the names $\mathcal{L}_1$ and $\mathcal{L}_2$ of the elements in two different ontology versions V_1 and V_2. Suppose also that there are two names of elements $L_i \in \mathcal{L}_1$ and $L_j \in \mathcal{L}_2$. The Linguistic Similarity between L_1 and L_2 is:

$$LS(L_i, L_j) := \max(0, \frac{\min(|L_i|, |L_j|) - ed(L_i, L_j)}{\max(|L_i|, |L_j|)}) \in [0,1].$$ LS returns a degree of

similarity between 0 and 1, where 1 means perfect map and 0 means worst map.

The structural mapping compares the structures of the elements in the taxonomies of different ontology versions and computes the structural similarities between them.

Suppose that there are two elements E_1 and E_2 from two different hierarchies H_1 and H_2 in different ontology versions. The set $N(E_i, H)$ includes all the neighbour elements of E_i in H.

$$N(E_i, H) := \{E_j \in E \mid H(E_j, E_i) \vee H(E_i, E_j)\}.$$ The set $Anc(E_i, H)$ includes all the ancestor elements of E_i in H.

The Structure Similarity (SS) between H_1 and H_2 as seen from the nodes may then be computed by following.

$$SS(E_1, H_1, E_2, H_2) = \frac{|(N(E_1, H_1) \cup Anc(E_1, H_1)) \cap (N(E_1, H_1) \cup Anc(E_1, H_1))|}{|(N(E_1, H_1) \cup Anc(E_1, H_1)) \cup (N(E_1, H_1) \cup Anc(E_1, H_1))|}$$

These mapping algorithms are based on Maedche's algorithm [9] and do some modifications to fit for ontology versioning.

From *LS matrix* and *SS matrix*, the weighted similarity *WS* is computed. The weighted similarity (*WS*) is a mean of *LS* and *SS*: *WS = w× SS + (1-w) × LS*, where *w* is the weight value in the range 0 to 1. Mapping elements are generated using the computed linguistic and structural similarities. A threshold th_{accept} can be set to identify the mappings. If ($WS(s, t) \geq th_{accept}$), then a mapping element from *s* to *t* is re-

turned. The values of the weighted value w and the threshold th_{accept} can be decided by the user or suggested by the system.

4   Related Work

Ontoview is a web-based system that provides support for the versioning of online ontologies [5]. Ontoview classifies different types of differences by highlighting them in different colours. PROMPTDiff compares different ontology versions and relate similar elements of them [6]. PROMPTDiff uses a set of heuristic matchers, each of which can deal with a specific mapping occasion and a fixed-point algorithm to combine the results of the matchers to produce a structural mapping between two versions. MRAFA is an Ontology MApping FRAmework (MAFRA) for distributed ontologies in the semantic web [9]. MAFRA uses semantic bridges to generate mappings between ontologies elements. ONION (ONtology compositION) system is an architecture based on a sound formalism to support a scalable framework for ontology integration [10]. ONION uses articulations of ontologies to interoperate among ontologies and uses articulation ontology to describe the mappings between two ontologies. GLUE is an ontology mapping system that uses machine-learning techniques to semi-automatically create semantic mappings between ontologies [11]. It computes the similarities between concepts in different ontologies based on a multi-strategy learning approach. KRAFT is an agent architecture for the integration of heterogeneous information systems [12]. KRAFT uses "ontology clustering" to integrate the heterogeneous ontologies. Table 1 describes the comparison of the features of these mapping approaches.

Table 1. Comparison of Single-directional Mapping Approaches

	Ontoview	PromptDiff	MAFRA	ONION	GLUE	KRAFT
Ontologies Mapping			√	√	√	√
Ontology Versioning	√	√				
M:N mapping			√	√		√
Linguistic level Mapping	√	√	√	√	√	√
Structure level Mapping			√	√	√	
Similarity computation			√	√	√	
Automatic		√				
Semi-automatic	√		√	√	√	√

Ontology mapping maps different ontologies and ontology versioning maps different ontology versions. In these approaches, only MAFRA, ONION and KRAFT can generate m: n mappings. The linguistic mapping identifies similarities between labels of the elements of the different ontologies or ontology versions. The structure mapping identifies the similarity between the taxonomies of elements in different ontologies or ontology versions. Similarity-based technique uses some matching algorithms

to compute the similarity between elements from ontological resources and identify the mappings based on the similarities. The rule-based technique uses some rules and constraints to identify the mappings. Most of these approaches require some metadata for generating the mapping that are provided by the user, which means that they are all semi-automatic. There is only one approach PromptDiff that is fully automatic.

5   Conclusion

This paper introduced a new type of ontology versioning: bi-directional ontology version (BOV), which can map two ontology versions bi-directionally at the same time. This paper gave the formal definition of BOV and described the components of it. The innovation of BOV lies in the bi-directional character. BOV can relate and transform similar elements of different ontology versions in both directions. This will improve the ability of data sharing and reuse among distributed data resources.

There are still some limitations of BOV. When there are many synonyms and homonyms, the accuracy of the mapping will be affected. The mapping algorithms can not identify the synonyms and homonyms by themselves alone. Some other technologies such as WordNet can help to solve this problem. BOV only supports 1:1 mapping if there is no transformation metadata provided by the user. So some single-directional only multiple candidates mapping will be lost in the bi-directional mapping. And that would induce losing some related knowledge between the elements in different ontology versions.

References

1. Gruber, T. R. A translation approach to portable ontology specifications. Knowledge Acquisition (1993).
2. Guarino, N.: Formal ontology, conceptual analysis and knowledge representation. International Journal of Human and Computer Studies, 43(5-6): 625-640 (1995)
3. Gray, P.M.D., Preece, A., Fiddian, N. J., Gray, W.A., Bench-Capon, T.J.M., Shave, M.J.R., Azarmi, N., and Wiegand, M.: KRAFT: Knowledge fusion from distributed databases and know ledge bases. In R.R. Wagner, editor, Eighth International Workshop on Database and Expert System Applications (DEXA-97), pages 682—691 (1997).
4. Klein, M. and Fensel, D.: Ontology versioning for the Semantic Web. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, pages 75-91, Stanford University, California, USA, July~30 - August~1 (2001).
5. Klein, M., Fensel, D., Kiryakov, A. and Ognyanov, D.: Ontoview: Comparing and versioning ontologies. In *Collected Posters ISWC 2002*, Sardinia, Italy (2002).
6. Noy, N. F and Musen, M. A.: PromptDiff: A Fixed-Point Algorithm for Comparing Ontology Versions. *In the Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Edmonton, Alberta (2002)
7. Zhao, Siyang and Tierney, Brendan: Bi-Directional Mapping between Ontology Versions: A Requirement for Data Sharing in Distributed Environments, Second International Conference on Computer Science and its Applications (ICCSA-2004), San Diego, California, USA, June 28-30, (2004).

8. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Dokl., (1966).
9. Maedche, A., Motik, B., Silva, N., Volz, R: MAFRA—A Mapping FRAmework for distributed ontologies. *Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW*, Madrid, Spain, (2002).
10. Mitra, P., Wiederhold, G., and Kersten, M.: A Graph-Oriented Model for Articulation of Ontology Interdependencies. *In Proceedings Conference on Extending Database Technology 2000 (EDBT'2000)*, Konstanz, Germany (2000).
11. Doan, A. H., Madhavan, J., Donfingos, P., Halevy, A. Y. :Learning to Map between Ontologies on the Semantic Web. In *Proc, of the Int. WWW Conference,* (2002).
12. Gray, P.M.D., Preece, A., Fiddian, N. J., Gray, W.A., Bench-Capon, T.J.M., Shave, M.J.R., Azarmi, N., and Wiegand, M.: KRAFT: Knowledge fusion from distributed databases and know ledge bases. In R.R. Wagner, editor, Eighth International Workshop on Database and Expert System Applications (DEXA-97), pages 682—691 (1997).