Mapping Refinement based on Ontology Evolution: A Context-Matching Approach

Julio Cesar dos Reis[1]

[1]Institute of Computing, University of Campinas (UNICAMP)
13083-852, Campinas, SP, Brazil

jreis@ic.unicamp.br

Abstract. *Ontologies and their associated mappings play a central role in several semantic-enabled tasks. However, the continuous evolution of ontologies requires updating existing concept alignments. Whereas mapping maintenance techniques have mostly handled revision and removal type of ontology changes, the addition of concepts and attributes demand further studies. This article proposes techniques to refine set of established mappings based on the evolution of ontologies. We investigate ways of suggesting correspondences with the new version of the ontology without applying a matching operation to the whole set of ontology entities. Obtained results explore the neighbourhood of concepts in the alignment process to update mapping sets.*

1. Introduction

Over the last decade several domains have exploited ontologies and their capabilities for various purposes ranging from information retrieval to data management and sharing. However, the size of these domains often requires the use of several ontologies whose elements are linked through mappings. Mappings are the materialization of semantic relations between elements of interrelated ontologies [Shvaiko and Euzenat 2013].

Creating mappings between ontologies is a complex task especially due to the increasing size of ontologies. To this end, several automatic ontology alignment techniques have been proposed [Mascardi et al. 2010]. Nevertheless, significant manual efforts of validation are still demanded if a certain level of quality is required [Shvaiko and Euzenat 2013]. This problem prevents software applications relying on mappings to fully take advantage on them.

Ontologies evolve over time to keep them up-to-date according to the domain knowledge. Ontology changes may affect mappings already established or can be a source for treating mapping refinement. In this context, in order to avoid the costly ontology re-alignment process, it is crucial to have adequate mapping adaptation strategies to keep mappings semantically valid over time [Reis et al. 2015]. Manual mapping maintenance is possible only if modifications are applied to a restricted number of mappings. Otherwise automatic methods are required for large and highly dynamic ontologies. For example, biomedical ontologies usually contain hundred of thousands of concepts interconnected via mappings.

Coping with the mapping reconciliation problem in a semi-automatic way entails many research challenges. First, it is difficult to evaluate the real impact of the ontology evolution on mappings. For instance, changing an attribute value may lead to invalidate

a mapping in some cases. In these situations, the challenging issue is to identify and classify the different cases. Second, several types of ontology changes can be applied to an ontology, but it is unknown how these different types of operations should be duly taken into consideration for mapping reconciliation [Groß et al. 2013]. Thus, generally only the removal of concepts is addressed. Third, the reconciliation of mappings with several types of semantic relations need deeper studies [Reis et al. 2015].

The design of techniques for mapping adaptation according to the different types of ontology changes has been coped within existing approaches. Previous work in literature [Reis et al. 2015] presented a mapping adaptation strategy for two out of three categories of ontology evolution: *removal* of knowledge and *revision* of knowledge. For example, when concepts are removed, heuristics were designed to automatically apply adaptation actions over mappings. The *addition* of knowledge (third category) is the most frequent type of change occurred in ontology evolution. New concepts or attributes in concepts are added to comply with new domain knowledge. Such new knowledge needs to be aligned with the interrelated ontologies, but we need to avoid applying a matching operation with the whole target ontology.

Literature presents several interpretations for ontology mapping refinement. The first interpretation is in the sense of helping to expand the types of semantic relations identified during the matching process. Refinement can modify or enrich semantic relations; for instance, during the refinement process, an equivalence ($\equiv$) relation (*i.e.*, a relationship defining that two interrelated concepts are equivalent) can be modified to an *is-a* ($\sqsubseteq$) relationship (*i.e.*, representing a relationship where one concept is a specialization of the other) [Arnold and Rahm 2014]. Another sense is refinement operations as a way of adding (deriving) or removing mappings, for instance, based on defined pattern rules [Hamdi et al. 2010]. These rules can be created manually or automatically, and are used to validate the mapped relations. The second category aims to enrich relations in a previously calculated mapping and the mapping set with new derived mappings. In this paper, we explore the second option.

In this paper, we propose a mapping refinement methodology to update mapping sets taking ontology changes into account (based on new concepts added in ontology evolution). We study the use of conceptual information related to neighbour concepts for enhancing the mapping adaptation quality. For this purpose, we investigate techniques to reuse already established mappings and to explore the role of neighbour concepts to derive new mappings. Our proposal allows suggesting new correspondences without applying a matching operation with the whole set of ontology entities. The remainder of this article is organized as follows: Section 2 presents the formal definitions and problem statement. Section 3 reports on our approaches to refine ontology mappings under ontology evolution. Section 4 draws conclusions and future work.

2. Preliminaries

Ontology. An ontology $\mathcal{O}$ specifies a conceptualization of a domain in terms of concepts, attributes and relationships [Gruber 1993]. Formally, an ontology $\mathcal{O} = (\mathcal{C}_{\mathcal{O}}, \mathcal{R}_{\mathcal{O}}, \mathcal{A}_{\mathcal{O}})$ consists of a set of concepts $\mathcal{C}_{\mathcal{O}}$ interrelated by directed relationships $\mathcal{R}_{\mathcal{O}}$. Each concept $c \in \mathcal{C}_{\mathcal{O}}$ has a unique identifier and is associated with a set of attributes $\mathcal{A}_{\mathcal{O}}(c) = \{a_1, a_2, ..., a_p\}$. Each relationship $r(c_1, c_2) \in \mathcal{R}_{\mathcal{O}}$ is typically a triple (c_1, c_2, t)

where t is the relationship (*e.g.*, "is_a", "part_of", "adviced_by", *etc.*) interrelating c_1 and c_2.

Context of a concept. We define the context of a particular concept $c_i \in \mathcal{C}_\mathcal{O}$ as a set of *super concepts*, *sub concepts* and *sibling concepts* of c_i, as following:

$$CT(c_i, \lambda) = sup(c_i, \lambda) \cup sub(c_i, \lambda) \cup sib(c_i, \lambda) \tag{1}$$

where

$$
\begin{array}{rcl}
sup(c_i, \lambda) & = & \{c_j | c_j \in \mathcal{C}_\mathcal{O}, r(c_i, c_j) = \text{``} \sqsubset \text{''} \wedge length(c_i, c_j) \leq \lambda \wedge c_i \neq c_j\} \\
sub(c_i, \lambda) & = & \{c_j | c_j \in \mathcal{C}_\mathcal{O}, r(c_j, c_i) = \text{``} \sqsubset \text{''} \wedge length(c_i, c_j) \leq \lambda \wedge c_i \neq c_j\} \\
sib(c_i, \lambda) & = & \{c_j | c_j \in \mathcal{C}_\mathcal{O}, sup(c_j) = sup(c_i) \wedge length(c_i, c_j) \leq \lambda \wedge c_i \neq c_j\}
\end{array} \tag{2}
$$

where λ is the length between two concepts (in terms of their relationship distance in the hierarchy of concepts) and the "$\sqsubset$" symbol indicates that "c_i is a sub concept of c_j". This definition of $CT(c_i, \lambda)$ is specially designed as the relevant concepts to be taken into account in the settings of this investigation on mapping refinement.

Similarity between concepts. Given two particular concepts c_i and c_j, the similarity between them can be defined as the maximum similarity between each couple of attributes from c_i and c_j. Formally:

$$sim(c_i, c_j) = \arg\max \; sim(a_{ix}, a_{jy}) \tag{3}$$

where $sim(a_{ix}, a_{jy})$ is the similarity between two attributes a_{ix} and a_{jy} denoting concepts c_i and c_j, respectively. We can compute this similarity at different linguistic levels: *character*, *string* and *semantic* level [Dinh et al. 2014].

Mapping. Given two concepts c_s and c_t from two different ontologies, a mapping m_{st} can be defined as:

$$m_{st} = (c_s, c_t, semType, conf) \tag{4}$$

where $semType$ is the semantic relation connecting c_s and c_t. In this article, we differentiate *relation* from *relationship*, where the former belongs to a mapping and the later to an ontology. The following types of semantic relation are considered: *unmappable* [$\perp$], *equivalent* [$\equiv$], *narrow-to-broad* [$\leq$], *broad-to-narrow* [$\geq$] and *overlapped* [$\approx$]. For example, concepts can be equivalent (*e.g.*, "head"$\equiv$"head"), one concept can be less or more general than the other (*e.g.*, "thumb"$\leq$"finger") or concepts can be somehow semantically related ($\approx$). The $conf$ is the similarity between c_s and c_t indicating the confidence of their relation [Euzenat and Shvaiko 2007]. We define $\mathcal{M}_{ST}^j$ as a set of mappings m_{st} between ontologies $\mathcal{O}_S$ and $\mathcal{O}_T$ at a given time j. We assume $j \in N$ the version of the ontology release $\mathcal{O}_S^j$. Therefore, ontology $\mathcal{O}_S^0$ is the version 0 whereas $\mathcal{O}_S^1$ is the version 1 of the same ontology.

     Ontology change operations (OCO). An ontology change operation (OCO) is defined to represent a change in an attribute, in a set of one or more concepts or in a relationship between concepts. OCOs are classified into two main categories: *atomic* and *complex* changes. Each OCO in the former cannot be divided into smaller operations while each one of the latter is composed of more than one atomic operation. For instance, the operation $chgA(c, a, v)$ is composed of two atomic operations $delA(a, c)$ and $addA(a, c)$. In this paper, we pay further attention to the operations of concept addition.

Problem statement. Consider two versions of the same source ontology $\mathcal{O}_S^j$ at time j and $\mathcal{O}_S^{j+1}$ at time $j+1$, a target ontology $\mathcal{O}_T^j$, and an initial set of mappings $\mathcal{M}_{ST}^j$ between $\mathcal{O}_S^j$ and $\mathcal{O}_T^j$ at time t_j. Suppose that the frequency of new releases of $\mathcal{O}_S$ and $\mathcal{O}_T$ is different and at time $j+1$ only $\mathcal{O}_S$ evolves. Since this evolution is likely to impact the mappings $\mathcal{M}_{ST}^j$, it is necessary to refine $\mathcal{M}_{ST}^j$ to guarantee the quality and completeness of $\mathcal{M}_{ST}^{j+1}$. The quality is related to the consistency of mappings. For instance, mappings cannot be established between removed concepts. The completeness refers to the recall of aligned concepts in $\mathcal{M}_{ST}^{j+1}$. In this investigation, we study how $\mathcal{M}_{ST}^j$ can be refined (e.g., new mappings derived) based on ontology changes related to *addition of knowledge*. In particular, we address the following research questions:

- How to exploit existing mappings for mapping refinement?
- Is it possible to reach mapping refinement for alignment of new concepts without applying a matching operation in the whole target ontology?
- What is the impact of using the context of concepts $CT(c_i, \lambda)$ in the target ontology on the mapping refinement effectiveness?

We consider that $\mathcal{O}_T$ has not evolved (thus $\mathcal{O}_T^j$ and $\mathcal{O}_T^{j+1}$ are the same version of the ontology $\mathcal{O}_T$). $\mathcal{O}_S^j$ and $\mathcal{O}_S^{j+1}$ are two distinct versions of the same ontology $\mathcal{O}_S$. At time $j+1$, newly added concepts appear in $\mathcal{O}_S^{j+1}$ and we attempt to refine the original mapping set M_{ST}^j to provide a set of valid mappings M_{ST}^{j+1}.

3. Mapping Refinement under Concept Addition Changes

At the first step, our approach identifies all newly added concepts using the *Conto-Diff* tool [Hartung et al. 2013]. This tool allows to identify atomic and complex ontology changes. Next, we extract the contextual information, *i.e.*, super, sub and sibling concepts of those newly added concepts (*cf.* Formula 1). At this stage, we consider $\lambda = 1$ (*i.e.*, direct parents and children concepts). We then examine the existing mappings between the source concept in the context of the newly added concept and the corresponding target concepts.

We propose adequate correspondences for each newly added concept. The idea behind the context-oriented technique is that the candidate mapping is established between a newly added concept and a target concept of an existing mapping at time j. Figure 1 illustrates the general scenario for the local approach to mapping refinement where we consider that $\mathcal{O}_T^j$ and $\mathcal{O}_T^{j+1}$ are the same ontology while the source ontology $\mathcal{O}_S^j$ evolved to $\mathcal{O}_S^{j+1}$ and presents new added concepts at time $j+1$.

Algorithm 1 computes the *diff* between two given versions of the source ontology (line 1). For each newly added concept c_i^1, the algorithm considers a candidate concept namely c_t in the target ontology by exploiting already existing mappings related to $CT(c_i^1, 1)$ (lines 4-8). We determine a new refined mapping by calculating the similarity between a new concept c_i^1 of $\mathcal{O}_S^{j+1}$ and a candidate c_t. If the maximum similarity (among the concept attributes) is greater than or equal to a threshold τ, the algorithm establishes a mapping between the newly added concept and the candidate target concept. Algorithm 1 searches for the candidate c_t that yields the maximum similarity value. The semantic type of the new mapping is computed according the relationship between the newly added concept and the optimal concept $c_{ij}^1 \in CT(c_i^1)$. For instance, if $c_{ij}^1 \in sup(c_i^1)$ so the $m_{it}.sem_type \leftarrow \leq$. This aspect requires further research to determine the type of semantic relation of the derived new mapping.
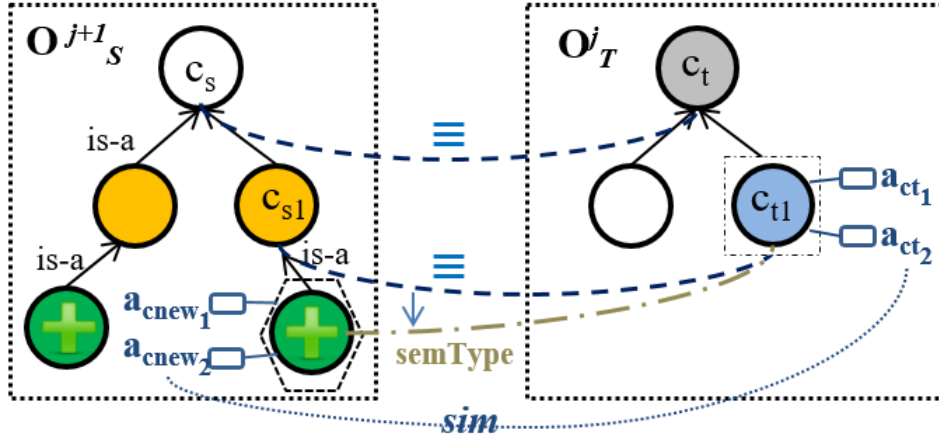
Figura 1. Context approach for mapping refinement.

Algorithm 1: Contextual approach to mapping refinement

Input: $\mathcal{O}_S^j, \mathcal{O}_S^{j+1}, \mathcal{O}_T^j, \mathcal{O}_T^{j+1}, \mathcal{M}_{ST}^j, \tau \in \mathbb{R}$

Output: $M_A = \{m_1, m_2, ..., m_N\}$

1:   $\mathcal{C}_{add} \leftarrow diff_{add}(\mathcal{O}_S^j, \mathcal{O}_S^{j+1})$ {newly added concepts}
2:   $\mathcal{C}_t \leftarrow \emptyset$ {initialize target concepts of candidate mappings}
3:   **for all** $c_i^1 \in \mathcal{C}_{add}$ **do**
4:     **for all** $c_{ij}^1 \in CT(c_i^1, 1)$ **do**
5:       **if** $\exists c_t^0 \in \mathcal{C}_{\mathcal{O}_T^0}, \exists m(c_{ij}^0, c_t^0) \in \mathcal{M}_{ST}^0$ **then**
6:         $\mathcal{C}_t \leftarrow \mathcal{C}_t \bigcup \{c_t^0\}$
7:       **end if**
8:     **end for**
9:     $m_{it} \leftarrow \emptyset$
10:    **for all** $c_t \in \mathcal{C}_t$ **do**
11:      $m_{cand} \leftarrow \text{argmax } sim(c_i^1, c_t)$ {Create a mapping between concepts c_i^1 and c_t}
12:      **if** $\max(sim(c_i^1, c_t)) \geq \tau$ **then**
13:        $m_{it} \leftarrow m_{cand}$
14:        $\tau \leftarrow \max(sim(c_i^1, c_t))$
15:      **end if**
16:     **end for**
17:     $\mathcal{M}_A \leftarrow \mathcal{M}_A \bigcup \{m_{it}\}$
18: **end for**
19: **return** $\mathcal{M}_A$

---

4. Conclusion

Ontology mappings play a central role for semantic data integration. However, domain knowledge update leads to new concepts in ontology versions. This requires to maintain mapping sets up-to-date according to the knowledge dynamics. In this article, we proposed techniques to refine ontology alignments based on evolving ontologies. We defined an algorithm to create contextual alignment to increase the chances of obtaining matching between concepts. Our developed approach relied on similarity computed between

concept attributes to maximize the similarity value (semantic confidence) between the interrelated mappings proposed. Future work involves further investigating heuristics to update the type of semantic relation in the refinement in the investigated procedures. We plan to conduct exhaustive experiments in different domains to evaluate the quality of detected refined mappings.

Acknowledgements

Referências

[Arnold and Rahm 2014] Arnold, P. and Rahm, E. (2014). Enriching ontology mappings with semantic relations. *Data & Knowledge Engineering*, 93:1–18.

[Dinh et al. 2014] Dinh, D., Reis, J. C. D., Pruski, C., Silveira, M. D., and Reynaud-Delaitre, C. (2014). Identifying relevant concept attributes to support mapping maintenance under ontology evolution. *Web Semantics*, 29:53 – 66.

[Euzenat and Shvaiko 2007] Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*. Springer.

[Groß et al. 2013] Groß, A., Reis, J. C. D., Hartung, M., Pruski, C., and Rahm, E. (2013). Semi-Automatic Adaptation of Mappings between Life Science Ontologies. In *Proceedings The 9th International Conference on Data Integration in the Life Sciences*, pages 90–104.

[Gruber 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220.

[Hamdi et al. 2010] Hamdi, F., Reynaud, C., and Safar, B. (2010). Pattern-based mapping refinement. *Knowledge Engineering and Management by the Masses*, pages 1–15.

[Hartung et al. 2013] Hartung, M., Gross, A., and Rahm, E. (2013). COnto-Diff: Generation of Complex Evolution Mappings for Life Science Ontologies. *Journal of Biomedical Informatics*, 46:15–32.

[Mascardi et al. 2010] Mascardi, V., Locoro, A., and Rosso, P. (2010). Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 22(5):609–623.

[Reis et al. 2015] Reis, J. C. D., Pruski, C., Silveira, M. D., and Reynaud-Delaitre, C. (2015). Dykosmap: A framework for mapping adaptation between biomedical knowledge organization systems. *Journal of Biomedical Informatics*, 55:153 – 173.

[Shvaiko and Euzenat 2013] Shvaiko, P. and Euzenat, J. (2013). Ontology Matching: State of the Art and Future Challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176.