A Reasoning-Based Support Tool for Ontology Mapping Evaluation

Christian Meilicke[1], Heiner Stuckenschmidt[1], Ondřej Šváb-Zamazal[2]

[1]University of Mannheim, {christian, heiner}@informatik.uni-mannheim.de
[2]University of Economics, Prague, ondrej.zamazal@vse.cz

Abstract. In this paper we describe a web-based tool that supports the human in revising ontology alignments. Our tool uses logical reasoning as a basis for detecting conflicts in mappings and exploits these conflicts to propagate user decision. The proposed approach reduces the effort of the human expert and points to logical problems that are hard to find without support.

1 Motivation

The alignment of ontologies is a common problem on the semantic web as many tasks such as information integration or semantic search rely on integrated background knowledge. As manual ontology alignment is a difficult and time-consuming process, a variety of algorithms and systems for computing matches between elements of different ontologies have been developed [2]. Almost all of these methods rely on linguistic or structural heuristics for deciding whether to regard elements from different ontologies as equivalent or not. These heuristics are bound to fail in many situations resulting in erroneous mappings that need to be corrected manually. As it turns out, this manual correction is a very difficult task that requires a good understanding of and sufficient information about the ontologies to be aligned. Motivated by this experience gained from five years of ontology alignment benchmarking carried out in the context of the Ontology Alignment Evaluation Initiative (OAEI) [1] we developed a web-based tool that supports the human user when evaluating a mapping. The distinguishing feature of this tool is the use of logical reasoning as a basis for detecting conflicts in mappings and implications of evaluation decisions. As shown in previous work [5] this approach can significantly reduce the effort of a manual evaluation and as argued in [4] also provides the basis for improving the correctness of mappings. In the following we first briefly discuss the kind of reasoning performed by our system. We then describe our system and present a typical usage example. We close with a brief description of the systems used in the context of the OAEI and some discussion of future work.[1]

[1] A demo version applied to the use case of Section 4 can be found at http://webrum.uni-mannheim.de/math/lski/ontdebug/mapdebug/debugpages/.

2 Reasoning-based Mapping Verification

Similar to Euzenat and Shvaiko [2] we define a mapping as a set of correspondences. A correspondence is a 4-tuple $\langle i\#E, j\#E', R, n\rangle$ where $i\#E$ and $j\#E'$ are concepts or properties of the matched ontologies $\mathcal{O}_i$ and $\mathcal{O}_j$, $R \in \{\sqsubseteq, \equiv, \sqsupseteq\}$ is a semantic relation, and n is a confidence value. In [5] we distinguished between two reasoning tasks useful for supporting manual mapping revision. On the one hand it can be decided whether a correspondence can be derived from a mapping, on the other hand it can be decided whether a mapping is incoherent.

We focused on the second aspect using a sound but complete reasoning technique that is based on two kinds of propagation patterns to detect incoherence conflicts between pairs of correspondences. We refer to these pattern as *subsumption propagation pattern* and *disjointness propagation pattern* both depicted in Figure 1. The subsumption propagation pattern works as follows. Suppose that a mapping $\mathcal{M}$ contains correspondences $\langle i\#A, j\#B, \sqsupseteq, n\rangle$ and $\langle i\#C, j\#D, \sqsubseteq, n'\rangle$, represented as arrows. Further suppose that $\mathcal{O}_i \models i\#A \sqsubseteq i\#C$. Due to the $\mathcal{M}$ it can be derived that $\mathcal{O}_i \cup \mathcal{M} \cup \mathcal{O}_j \models j\#B \sqsubseteq j\#D$ and also $\mathcal{O}_i \cup \mathcal{M} \cup \mathcal{O}_j \models j\#E \sqsubseteq j\#D$ for each subconcept $j\#E$ of $j\#B$. Suppose now that $j\#E$ and $j\#D$ are defined to be disjoint in $\mathcal{O}_j$. From this we can conclude that $j\#E$ has become an unsatisfiable concept due to the additional terminological axioms introduced by $\mathcal{M}$. Thus $\mathcal{M}$, respectively its subset $\{\langle i\#A, j\#B, \sqsupseteq, n\rangle, \langle i\#C, j\#D, \sqsubseteq, n'\rangle\}$ is incoherent. We have to remove at least one element to resolve the incoherence.

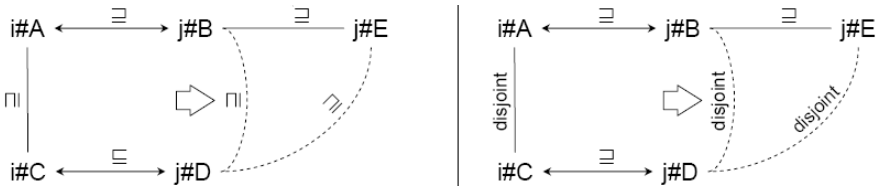

Fig. 1. Subsumption (on the left) and disjointness propagation pattern (on the right).

The disjointness propagation can be understood as the inverse of the subsumption propagation, thus we omit a detailed description. Once we classified both ontologies, the presence of these patterns has to be checked in both direction by a sequence of lookups for all possible pairs of correspondences. This process can be seen as a diagnosis of the incoherence. While we presented both patterns for subsumption correspondences, our approach is in the same way applicable for equivalence correspondences since such a correspondence can be decomposed into two subsumption correspondences. We apply the same approach to correspondences between properties in a similar way by exploiting their domain and range restrictions.

3 A Web Based Tool for Mapping Verification

When a mapping is loaded, the system first performs a diagnosis step according to the approach described in the previous section. In this step, conflicts between pairs of correspondences are identified. The system then presents a list of correspondences together with their confidence value as provided by the matching system. Depending on the status of the evaluation, each correspondence is labeled with one of the following symbols (see Figure 2 for an example):

✔ The correspondence has been *manually accepted*. It is assumed that this correspondence is correct and any conflicting correspondences are incorrect.

✔ The correspondence has *not yet been evaluated and does not conflict* with any accepted or not yet evaluated correspondence.

⚡ The correspondence *conflicts* with an accepted or not yet evaluated correspondence.

🗑 The correspondence has been *dismissed automatically* as it conflicts with a manually accepted correspondence.

🗑 The correspondence has been *manually dismissed* by the user. Any correspondence previously in conflict with this correspondence changes its status if there exists no other non-dismissed conflicting correspondence.

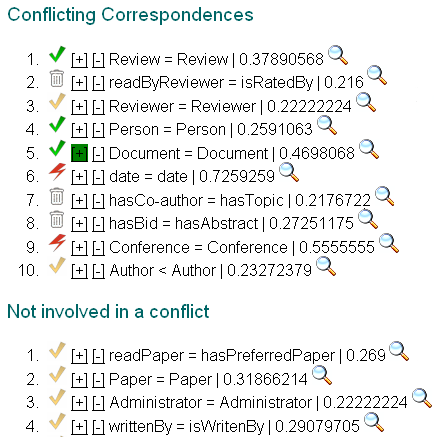

Fig. 2. User interface

In an ideal case, the evaluation is carried out until all correspondences are labeled with ✔, 🗑 or 🗑 indicating that it has been confirmed directly or indirectly by the human expert to be either correct or incorrect. In the case of large mappings, this will not always be the case as the user will focus on resolving conflicts. In this case, correspondences labeled with ✔ and ✔ are assumed to be correct and correspondences labeled with ⚡, 🗑 or 🗑 are assumed to be incorrect. Figure 3 shows a typical example of how the evaluation tool is used. The data used in the example is the result of the DSSim matching system on the CMT and CRS ontology both of which are part of the OntoFarm dataset [6]. This matcher generates twelve correspondences eight of which are involved in a conflict. These correspondences are shown in Figure 3(a) as presented to the user at the beginning of the evaluation process.

In a first step the user might check correspondence *Review* = *review*. The tool supports his decision by showing the context of the matched elements when clicking on the lens symbol. In the case of concepts, the path to the top of the hierarchy is presented for both concepts. Figure 3(b) shows the corresponding context for the first correspondence in the set. Based on this additional information the user will decide that the correspondence is correct and mark it as manually accepted by clicking on the (+) symbol changing its label from ⚡ to

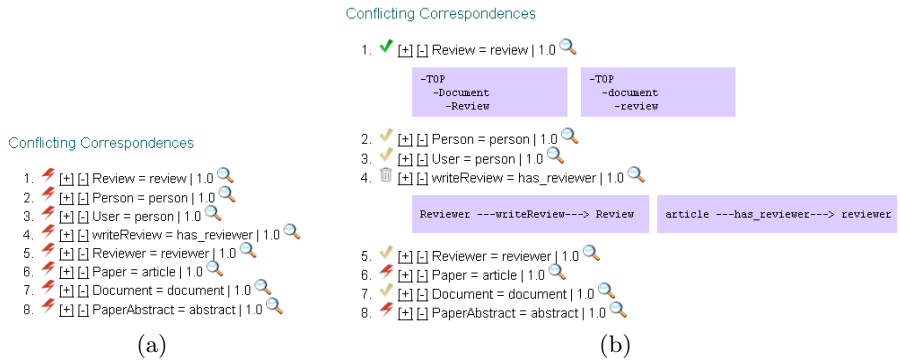

Fig. 3. Examples of an evaluation process

✔. This single decision triggers a number of events. In particular, correspondence 4 (*writeReview* = *has_reviewer*) can now be decided to be incorrect as it conflicted with correspondence 1 and will be relabeled as 🗑. As Figure 3(b) shows assuming that correspondence 4 is incorrect resolves a number of other conflicts which leads to a relabeling of correspondences 2, 3, 5 and 7 from ⚡ to ✓. The fact that the rejection of correspondence 4 resolves many other conflicts is already an indication that the decision should be correct. In order to be sure that the correspondence is actually incorrect, the user can additionally inspect the context information provided by the system which is shown in Figure 3(b). Looking at the context clearly shows that the two relations cannot be the same as they connect completely different concepts ('Reviewer' and 'Review' in the first and 'article' and 'reviewer' in the second case). In a similar way, the evaluation can be continued until all correspondences are correctly labeled.

4 Application in the OAEI Challenge

We applied our tool in the context of the OAEI conference track. This track deals with a collection of ontologies in the domain of conference organization known as the *OntoFarm* collection [6]. Since 2006 the evaluation procedure of the conference track the submitted mappings had been manually evaluated over the last years. This procedure resulted in a large corpus of annotated correspondences which can be seen as a first step towards building reference mappings. Since an enormous number of correspondences had to be processed, it can be expected that not every correspondence has been annotated correctly. Thus, we used the annotated corpus as input to our tool, revised it, and generated precise reference mappings over five ontologies (i.e. 10 mappings).

Using our tool we could explore conflicts between correspondences and track which correspondences would be dismissed provided another correspondence is evaluated as correct. This information turned out to be very useful in the revision process as we already argued with respect to the example presented above. It would have required an extensive and time consuming inspection of the involved

ontology to gain this information without support. We experienced that the best way to use the tool is to label the most certain correspondences at first. This resulted very often in automatically eliminating doubtful correspondences without the need for further investigation. Overall, we significantly reduced the input corpus of correspondences previously annotated as correct and could thus create a basis for highly precise reference mappings that will be used in future evaluations of the conference track.

5 Conclusions

There are different ontology editors that provide functionality for modeling and visualizing mappings such as Protege and OntoStudio. The support of these systems, however, is limited to some form of visualization of mapped elements and their context. Based on discussions with researchers in the area, we have identified a strong need for a more substantial support that we want to provide at least partially with our system. As the methods implemented in our system rely on logical reasoning, the system is particularly useful for evaluating mappings between richly axiomatized ontologies. However, as shown in previous work [3], it is also applicable to light-weight ontologies. Currently, the presentation of context information to the user is rather basic. In future work, we will enhance this feature. We plan to implement our tool as Protege-Plugin using Protege functionalities for visualizing the context of mapped elements.

References

1. C. Caracciolo, J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaise, C. Meilicke, J. Pane, P. Shvaiko, H. Stuckenschmidt, O. Svab-Zamazal and V. Svatek. Results of the Ontology Alignment Evaluation Initiative 2008. Proceedings of the ISWC 2008 Workshop on Ontology Matching, 2008.
2. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.
3. C. Meilicke, J. Voelker and H. Stuckenschmidt. Learning Disjointness for Debugging Mappings between Lightweight Ontologies. Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management.
4. C. Meilicke and H. Stuckenschmidt. Incoherence as a Basis for Measuring the Quality of Ontology Mappings. Proceedings of the ISWC 2008 Workshop on Ontology Matching, 2008.
5. C. Meilicke, H. Stuckenschmidt and A. Tamilin. Supporting Manual Mapping Revision using Logical Reasoning. Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, 2008.
6. O. Svab, V. Svatek, P. Berka, D. Rak, P. Tomasek. OntoFarm: Towards an Experimental Collection of Parallel Ontologies. Poster Track of the ISWC 2005.