Ontology mapping through analysis of model extension*

Xiaomeng Su, Terje Brasethvik and Sari Hakkarainen

Dept. of Computer and Information Science
Norwegian University of Science and Technology (NTNU)
N-7491 Trondheim, Norway
{xiaomeng, brase, sari}@idi.ntnu.no

Abstract. The overall problem addressed in this paper is to improve semantic interoperability in heterogeneous systems. Normally, the semantics of data is carried by ontology (concept model). Reconciling data semantics therefore boils down to reconciling relevant ontologies. A candidate solution is to use extensional information, i.e. the instance information of the ontology to enrich the ontology and further, based on the enrichment structure to calculate similarities between concepts in two ontologies.

1 Introduction

The overall problem addressed in this paper is to improve semantic interoperability in heterogeneous systems, where the semantics of data is carried by ontology. Thus the reconciliation of data semantics boils down to reconciling relevant ontologies.

One of the fundamental elements of the ontology integration process is establishing the mapping between ontologies. Mapping processes typically involve analysing the ontologies and comparing them in order to determine the correspondence among concepts and to detect possible conflicts. A set of mapping assertions is the main output of a mapping process. The mapping assertions can be used directly in a translator component, which translates statements that are formulated by different ontologies. Further, a follow-up integration process can use the mappings to conduct merging.

A fully automatic implementation of mapping is considered implausible and much of the research in this area is therefore focusing on semi-automating the mapping process. The approach introduced here is an instance of such research.

2 Model Extension analysis.

2.1 Overall approach.

To begin with, we are focusing on the so-called lightweight ontology. Hence, an *ontology* is defined as a set of elements connected by some structure. Among the structures, we single out hierarchical IS-A-relation and the others we call "related" rela-

* This work was partly supported by Accenture Norway.

tions, which is merely an indication of relatedness. A *classification hierarchy* is a typical example of ontology organized only by hierarchical IS-A-relation.

The first step in handling semantic heterogeneity should be the attempt to enrich the semantic information of concepts in ontologies, as it is well understood that the richer information the ontologies possesses, the higher probability that high quality mappings will be derived. Here, the extension, i.e. instance information of a concept is considered. The instances are documents that have been classified to the concepts. The intuition is that the written documents that are used in a domain inherently carry the conceptualizations that are shared by the members of the community.

The second step is to analyse correspondence relations among ontology elements. We consider information retrieval (IR) technique as one of the promising components of our approach. With information retrieval, a concept node in the first ontology is considered as a query to be matched against the collection of concept nodes in the second ontology. Ontology mapping thus becomes a question of finding concept nodes from the second ontology that best relate to the query node. Further, we adopt the correspondence assertion model in [3] as basis for describing the relevant information of mappings (Fig. 1)

Converging the above two ideas, it is evident that the instance information of a concept needs to be represented in a way that is compatible with IR framework. Given that vector space model is the most used one in IR, it is natural to think of representing the instance information in vectors, where the documents under one concept become building material for the vector of that concept. In some cases, ontologies exist without any available instance information. We tackle that by assigning relevant instance to the ontologies. That is where text categorization come into play, aiming at automate the process of assigning documents to concept nodes.

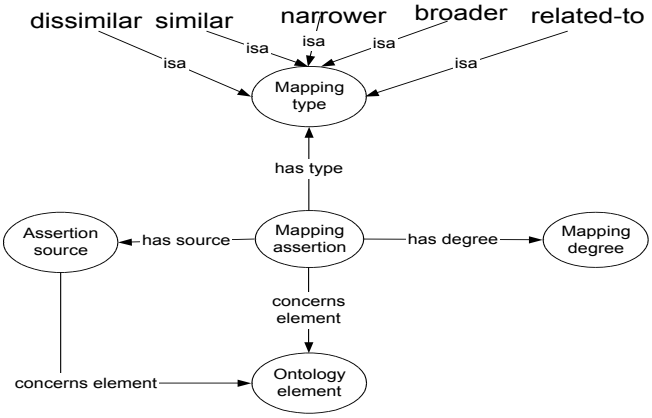

Fig. 1 mapping assertion metamodel

2.2 Mapping Discovery Method

Figure 2 depicts the general architecture of the suggested mapping process. The approach takes two ontologies and associated document set(s) as input. There can be one or two document sets. In the former case, we assume the documents are relevant to both ontologies, while in the latter, it is assumed that the two document sets share same vocabulary. Currently, parts of the system have been implemented; an example screenshot is depicted in fig 3. There are four steps in the process, as follows:

1. Text Categoriser assigns documents to the concepts nodes of the two ontologies respectively.
2. FVC (Feature Vector Constructor) builds up feature vectors for each concept nodes in the two ontologies.
3. Mapper calculates similarity value for each pair of the concept nodes
4. Mapping Refiner formulates mapping assertions, presents them to the user and manages the user feedback.

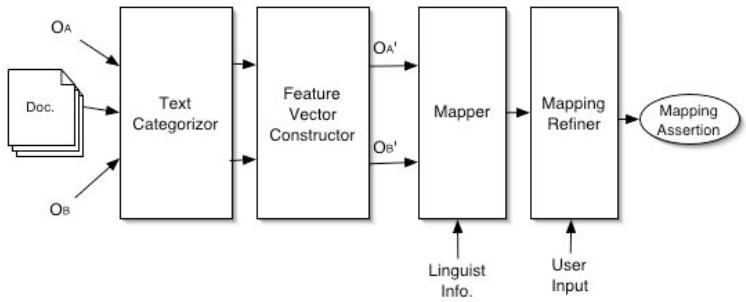

Fig. 2 Architecture of Ontology Mapping

The first step is to assign documents to concept nodes of the ontology. Text categorization technique is the first natural candidate for that task. The assigning of documents to concept nodes is necessary when no instance knowledge of the ontology is available. However, if documents have already been assigned to specific nodes, we proceed to the next step directly.

The second step concerns building up feature vectors for each concept node in the two ontologies. For each node a feature vector is calculated based on the associated document. Following a classic Rocchio algorithm [1], the feature vector for node a_i is computed as the average vector over all document vectors that belong to node a_i. Furthermore, the feature vector of a non-leaf node is computed as the centroid vector of its sub nodes. Thus, hierarchical information is partially taken into consideration. The output of this step is two intermediate ontologies, O_A' and O_B', where each concept node has been associated with a feature vector.

The third step produces initial mapping assertions based on the two intermediate ontologies. With the feature vector at hand, the similarity of the two nodes is measured by calculating the cosine measure of the two associated vectors. Further, supplementary information is used in order to acquire better mapping candidates. Here, we use linguistic information of names. A matching algorithm of class names is de-

ployed to give a boost for nodes, which have the same or similar names (prefix, suffix, or word root) with the compared one.

The final step involves presenting the candidate mapping assertions based on the similarity calculation to the user. Here the process also considers user feedback as to approve/reject the assertions and the management of the evolving mapping assertions.

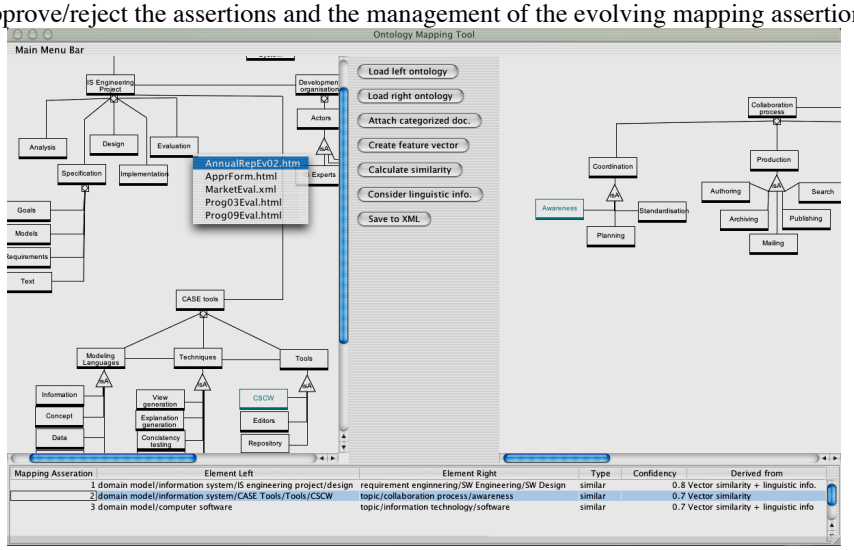

Fig. 3 A prototype of Ontology Mapping Tool

3 Conclusion and future work.

In this paper, we present a heuristic mapping method and a prototype mapping system that supports the process of semi-automatic ontology mapping. The mapping method has been inspired by both information retrieval and text categorisation techniques. The next logical step in our research is to gather empirical data about the performance and the applicability of the mapping system. To begin with, we conduct an experiment on the product catalogue task [2]. Second, our plan is to explore the performance in large real-world domains such as the KITH medical patient documents domain [4] and the library resource management domain.

References

1. Aas, K., Eikvil, L.: Text Categorisation: A Survey. Norwegian Computing Center, Oslo (1999)
2. Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A.: Product Data Integration for B2B E-Commerce. IEEE Intelligent Systems 16 (2001)
3. Hakkarainen, S.: Dynamic Aspects and Semantic Enrichment in Schema Comparison. PhD Thesis. Stockholm University (1999)
4. http://www.kith.no