SOMET: Shared Ontology Matching Environment

Liam Magee

Globalism Institute, School of Global Studies, Social Sciences and Planning
RMIT, Melbourne, Australia
liam@commonground.com.au

http://www.rmit.edu.au

Abstract. In this paper we present a tool, SOMET, for collaborative developing, matching and merging ontologies. The tool's design is based on a Wiki model, allowing for multiple authors to contribute to an ontology. It also provides a number of meta-ontology features, including the ability to compare, match and merge. The tool makes use of one algorithmic approach to element-level mapping, demonstrating the use of both automated and manual matching.

1 Introduction

Much research has been invested into automated techniques of ontology matching [6, 7]. There is general recognition of the need to augment these techniques with manual matching. *Collaborative* matching, utilising both automated and manual matches, is important to resolve conceptual ambiguities, and to promote re-use across organisational and geographical boundaries.

This paper presents an online tool, SOMET (pronounced 'Summit'), for collaborative developing, matching and merging ontologies. It is motivated by the idea that ontology development and matching is essentially a *social* and *interactive* activity, and is best served by tools which permit this. As such, the tool is based on the Wiki model for ontology development.

This paper is structured as follows: In Section 2, we review related work. Section 3 presents a case for collaborative ontology development and matching. Section 4 describes the design and implementation of the SOMET prototype, along with our approach to ontology matching and merging. Section 5 examines the test results of using SOMET on two sample ontologies. In Section 6, we look at further directions for SOMET. Section 7 concludes the paper.

2 Related Work

The concept of shared or collaborative ontology development is not new. A number of tools have been introduced, including CODE [5], KAON [3], OntoEdit [9], Ontolingua [4], WebODE [1] and Wiki@nt [2]. SOMET differs from these tools in focussing particularly on ontology *matching* and *merging* in a collaborative environment.

There has been considerable exploration of approaches for ontology matching. Much of the research has been into finding suitable algorithms to automate part or all of a given matching task. As shown by surveys [6, 7], such algorithms use a variety of strategies for matching ontologies. This paper explores the use and partial implementation of one such algorithm, S-MATCH [8]. The S-MATCH algorithm emphasises its semantic matching characteristics. However the algorithm also exploits syntactic and external techniques. As such, it is a good candidate for exploring the use of semi-automated techiques in a collaborative environment.

Recent work also suggests the importance of community-driven ontology matching [10]. This paper assumes development and matching of ontologies is often collaborative in nature, and requires tools such as SOMET to realise the benefit of community-driven domain models.

3 Overview of Collaborative Ontology Development and Matching

Ontologies are typically developed by a process of iterative *construction* and *consultation*, with a focus on concepts in a specified domain. For the most part construction and consultation are separate activities, conducted in serial fashion, as the modeller uses specific knowledge of the modelling environment to apply the results of the consultation process. In the case of traditional database and software engineering activities, such established practices are generally entrenched in broader lifecycle processes, and there has been little impetus to shift the onus of model construction from the modelling expert. For Semantic Web ontologies, where models are frequently shared across organisational and geographical boundaries, there is significantly greater motivation to develop such models collaboratively.

The approach used in this paper is based on the success of open access content systems. The approach accepts that a lower grade user interface will be acceptable in certain contexts, just as authoring content online is an acceptable degradation from using a word processor in certain contexts. In particular, if the ontology is small, then the frequent submission and retrieval of ontology elements across a network of ontology components will be tolerable.

4 Outline of SOMET System, and Matching and Merging Techniques

SOMET is a prototype that has been developed in Ruby, using the Ruby on Rails framework. It employs a commonly used model-view-controller architecture. It also employs a plug-in architecture for executing matching algorithms.

The SOMET interface makes it possible to construct an ontology with classes, properties and individuals. Notable features at this stage include the following:

- *Creation* and *editing* of ontologies, classes, properties, individuals and annotations.
- *Importing* and *exporting* an ontology.
- Generation of a *comparative report* of differences between two ontologies.
- Manual and semi-automated *matching* of classes.
- *Merging* of two ontologies.
- Various Wiki features, such as *publishing*, *sharing*, *publishing*, *versioning*, *logging* and *commenting* on ontologies. Class and property matches can be proposed, discussed, and approved or rejected.

We have conducted a partial implementation of the S-MATCH algorithm [8]. As there is not a suitable satisfiability engine in the Ruby language, we have not been able to implement Step 4 of the algorithm. Consequently we have not yet been able to test the *semantic* aspects of S-MATCH, which require the translation of the labels of the path of a given node on the ontology graph into a propositional logic formula. Instead we have translated steps 1-3 of S-MATCH to Ruby, using the OWL object model we have developed. We were able to develop a matrix of ontology labels with at least partial asssignments of the following relations: *equivalence* ($=$), *more general* ($\sqsupseteq$), *less general* ($\sqsubseteq$), and *disjointness* ($\perp$). The result is an element-based, *syntactic* and *external* technique, as outlined in [**?**] - but without the key *semantic* characteristics outlined in [8]. The following outlines the key steps of the algorithm, as presented in [8, 305-7]:

Step 1. For all labels L in the two trees, compute *concepts* of *labels*.

Step 2. For all nodes N in the two trees, compute *concepts* of *nodes*.

Step 3. For all pairs of labels in the two trees, compute *relations* among *concepts* of *labels*.

Step 4. For all pairs of nodes in the two trees, compute *relations* among *concepts* of *nodes*.

For our purposes, we utilised WordNet in steps 1 (to develop the senses of each lemma) and 3 (to generate the relations between pairs of labels). The generation of the label matrix at step 3 made use of an exhaustive traversal through the WordNet dictionary. The results of step 3 are a set of tuples, $\langle eID, n1\_i, n2\_j, R\rangle$, where *eID* is a unique identifier for the given element-level match, $n1_i$ is the i-th node of the first ontology graph, $n2_j$ is the j-th node of the second ontology graph, and R is one of equivalence, more general, less general or disjoint. We ignore possible overlapping or unknown relations. These tuples are captured in a database, grouped together as an ontology match.

The *Merge* operation generates a new ontology graph from two existing ones, on the basis of a defined ontology match. The resulting merged ontology is stored in the database. The operation has 5 steps:

1. Compute the set of direct *child-parent* relations for the new graph, based on the set of element-level matches.

2. Compute the set of direct *parent-child* relations for the new graph, based on the results of step 1.
3. Generate the complete set of parent and child nodes for the new graph, based both on relations from steps 1 and 2, as well as the existing relations in the source graphs.
4. Compute the set of disjoint relations for the new graph, based on the set of element-level matches, for all siblings in the graph generated by step 3, where such siblings are not already disjoint.
5. Perform a deep copy of non-matched objects from each of the source ontologies into the new ontology.

5 Test Results

The matching and merging capabilities of SOMET have been tested using two simple ontologies. The goals of the test were to successfully invoke the S-MATCH implementation, generate a set of class-level matches, add or modify one such match, and to merge the two ontologies into a third on the basis of the matches. The test would be successful a) if the merged ontology contained the union of the two source ontology classes in a directed acyclic graph, with at least some successful matches, and b) if the *Match* and *Merge* operations execute in reasonable time. Tests were conducted on a P4 3.0GHz machine with 2GB of memory.

The matching operation took 74.407 seconds. Further analysis showed the majority of this time was due to the exhaustive scansion of the WordNet database. The merging operation took 5.703 seconds. The results show some inconsistencies of the WordNet associations, with certain anomalous subsumtion relations identified.

The results of the test show that for small ontologies, the *Match* and *Merge* operations can be conducted in an shared online environment. While the performance is sub-optimal, this could be corrected by depth-limited WordNet searches, and augmented with domain-specific vocabularies. The quality of the match also varies, although implementing step 4 of the S-MATCH algorithm, ie. the satisfiability checks on formulas representing the full path of any given element, would improve this greatly. The test also demonstrates the ability to augment automated matching techniques with manual matching.

6 Further Work

SOMET has been developed to a prototype level, and as such lacks the kinds of user interface enhancements expected of such a tool in a production environment. To realise the aims of providing a generic matching tool with ‘pluggable’ matching techniques, a more sophisticated plug-in architecture needs to be developed. Given the diversity of matching approaches, the viability of this aim also needs to be better ascertained.

7 Conclusions

Research on publishing and document standards over several years has motivated the investigation into how ontologies can be developed and matched in a collaborative way. We have concluded *social interaction*, *negotiation* and *collaboration* are necessary aspects to successful ontology matching in many environments. So far, tools for ontology matching have focussed on *private* ontology matching. In this paper we have presented SOMET as a prototype for *collaborative* ontology development, matching and merging. The test results have been encouraging in terms of its utility for these purposes.

Acknowledgments. This work has been supported by ARC Grant LP0667834, 'Towards the Semantic Web: Standards and Interoperability across Document Management and Publishing Supply Chains'.

References

1. J. C. Arpirez, O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. Webode: a scalable workbench for ontological engineering. In *Proceedings of the international conference on Knowledge capture*, pages 6-13. ACM Press, 2001.
2. A. Bao, V. Honavar. Collaborative Ontology Building with Wiki@nt - A Multi-agent Based Ontology Building Environment. In *Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools (ISWC 2004)*. Technical Report, Computer Science, Iowa State University.
3. E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias. Kaon - towards a large scale semantic web. In *K. Bauknecht, A. M. Tjoa, and G. Quirchmayr, editors, E-Commerce and Web Technologies, Third International Conference, EC-Web 2002, Aix-en-Provence, France, September 2-6, 2002, Proceedings, volume 2455 of Lecture Notes in Computer Science, pages 304-313*. Springer, 2002.
4. A. Farquhar, R. Fikes, W. Pratt, and J. Rice. Collaborative ontology construction for information integration, 1995.
5. P. Hayes, R. Saavedra, and T. Reichherzer. A collaboration development environment for ontologies. In Proceedings of the Semantic Integration Workshop, Sanibel Island, Florida, 2003.
6. N. Noy. Semantic Integration. A Survey of Ontology-based Approaches. Sigmod Record, Special Issue on Semantic Integration, 2004.
7. P. Shvaiko, J. Euzenat. Survey of Schema-based Matching Approaches. Journal on Data Semantics, 2005.
8. P. Shvaiko, F. Giunchiglia, P. Pinheiro da Silva, D. L. McGuinness. Web Explanations for Semantic Heterogeneity Discovery. In Proceedings of ESWC, 2005.
9. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative ontology development for the semantic web. In *Proceedings of the First International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italia*. Springer, LNCS 2342, 2002.
10. A. Zhdanova, P. Shvaiko. Community-Driven Ontology Matching. In Proceedings of ESWC, 2006.