Ontology Evolution: A Practical Approach

Fouad Zablith*

Knowledge Media Institute (KMi), The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
f.zablith@open.ac.uk

Ontology evolution is increasingly getting research momentum in the Semantic Web field. This is due to the fact that ontologies, forming the backbone of Semantic Web systems, need to be kept up-to-date for ontology-based systems to remain usable. We highlight two research approaches in the domain of ontology evolution: The first considers the evolution as a pure management of changes performed by the user [7, 9–11], while the second takes into account dynamically updating and learning ontologies without offering extensive change and evolution management functionalities [1, 2, 8]. Many definitions of ontology evolution exist [5]. We understand ontology evolution as the “timely adaptation of an ontology to the arisen changes and the consistent management of these changes” [6]. This definition indirectly reflects the need of combining the two aforementioned approaches for achieving a successful evolution. Yet no practical and complete solutions exist that cover all stages of evolution.

We are planning to close the above gap by proposing a complete ontology evolution framework, Evolva[1] that: firstly covers the entire evolution cycle, and secondly makes use of background knowledge to potentially decrease, or even eliminate, user involvement. The need for Evolva emerged from the tedious and time consuming update and evolution of our KMi Semantic Web portal[2] ontology. Being highly user dependent and occurring in a dynamic domain, the ontology was left outdated. In this abstract we focus on the implementation of Evolva as part of the NeOn Toolkit[3], a novel ontology management framework. Figure 1 illustrates a screenshot of Evolva’s pilot system.

Evolva detects the need for evolution by contrasting the content of the ontology to evolve (i.e. base ontology appearing in the left panel of Figure 1), with the content of external data sources. Such data sources can consist of text documents, databases, folksonomies, or even other ontologies, and can be selected in the “Data Sources” panel. Evolva processes the sources in its information discovery component in order to extract ontological entities. Currently we focus on concepts, but will extend the system to deal with instances as well. The Text2Onto [4] extraction algorithms are used for processing text documents and identifying entities. The entities are then passed to the data validation step that selects new entities with respect to the base ontology by using a Jaro-based string matcher. During validation, automated methods remove noisy terms that, for ex-

* This work is funded by the NeOn project sponsored under EC grant IST-FF6-027595

[1] An overview of Evolva can be found in [12] and [13].

[2] http://semanticweb.kmi.open.ac.uk/

[3] http://www.neon-toolkit.org/

ample, fall below a minimum term length threshold. We also have a filter for removing irrelevant terms such as the generic ones (e.g. thing, individual). The user is able to interfere in the validation, and manually exclude entities he/she believes are irrelevant to the domain. This is done under the "Data Validation" panel.

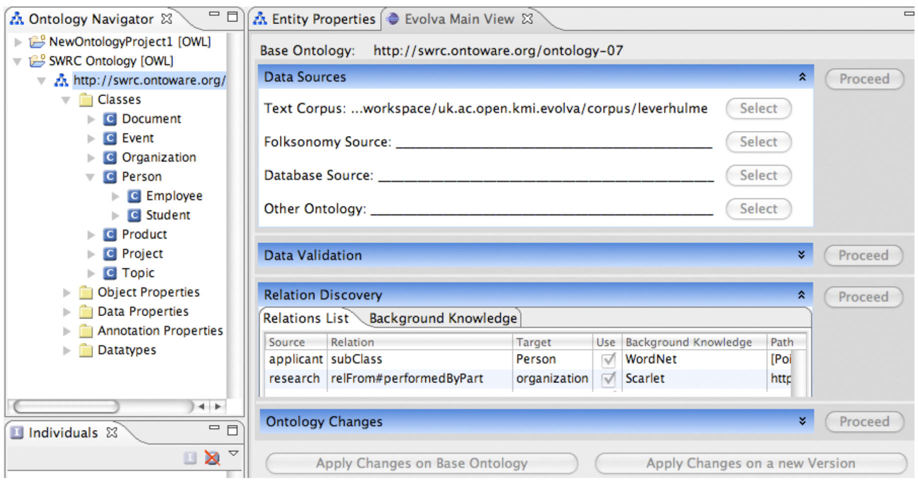

Fig. 1. Evolva Pilot System Screenshot

After the information discovery and validation stages, background knowledge is used for linking new and relevant entities to the base ontology. This is one of the core features of Evolva, as this stage is traditionally the most expensive in terms of user input. Background knowledge can be provided by different resources such as lexical databases, online ontologies and online documents. Our current implementation uses WordNet and online ontologies for relation discovery. WordNet contains hierarchy-based relations between terms and can be accessed quickly. Online ontologies are slower to access, but they offer a richer source of relations from a constantly increasing body of knowledge. We performed an experiment about the potential usage of such background knowledge sources for relation discovery, and they proved to have a high precision of around 77% [13]. Online ontologies are exploited using Scarlet[4], a relation discovery tool on the Semantic Web, from which hierarchy as well as named relations can be discovered. The "Relation Discovery" panel displays the *Source*, which is the new term extracted from the data sources, and its *Relation* to the *Target* term of the base ontology. Details of the relations such as the *Background Knowledge* used to discover it and its complete *Path* are also available. Figure 1 shows an example of how WordNet helped linking the new concept *Applicant* as a *subClassOf Person* (a concept in the base ontology). A second example shows how Scarlet

[4] http://scarlet.open.ac.uk/

links *Research* to *Organization*, through a *performedByPart* relation. The challenge here is to efficiently validate the relations, prior to applying any changes on the base ontology. E.g. how to select the right synset in WordNet, or how to determine whether a relation discovered from online ontologies does not conflict with the existing knowledge of the base ontology? Currently we are relying on the web-based distance similarity measure [3] as a step to check the possibility of two terms being related, before performing relation discovery. Part of our future plans is to use other validation techniques such as the base ontology itself as a validator, as well as word sense disambiguation. In addition to these automated validation methods, the user can manually exclude irrelevant relations.

The next step is to apply the changes on the base ontology using the relevant discovered relations. The changes can be applied either directly on the base ontology, or on a new detached copy of the base ontology. Our future implementation phase focuses on the two remaining components of Evolva: (1) evolution validation for consistency and duplication checks that could have occurred as an evolution side effect, and (2) the evolution management for recording changes and handling change propagation to the dependent components such as applications, or imported and aligned ontologies.

References

1. H. Alani, S. Harris, and B. O'Neil. Winnowing ontologies based on application use. *Proceedings of ESWC*, 2006.
2. Stephan Bloehdorn, Peter Haase, York Sure, and Johanna Voelker. *Ontology Evolution*, pages 51–70. John Wiley & Sons, June 2006.
3. R. L. Cilibrasi and P. M. B. Vitányi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, pages 370–383, 2007.
4. P. Cimiano and J. Völker. Text2onto - a framework for ontology learning and data-driven change discovery. *Proceedings of NLDB'05*, pages 15–17, 2005.
5. G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, and G. Antoniou. Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(02):117–152, 2008.
6. P. Haase and L. Stojanovic. Consistent evolution of owl ontologies. *Proceedings of ESWC*, pages 182–197, 2005.
7. M. Klein. *Change Management for Distributed Ontologies*. PhD thesis, Vrije Universiteit in Amsterdam, 2004.
8. V. Novacek, L. Laera, and S. Handschuh. Semi-automatic integration of learned ontologies into a collaborative framework. *IWOD*, 2007.
9. N. F. Noy, A. Chugh, W. Liu, and M. A. Musen. A framework for ontology evolution in collaborative environments. *Proc. of ISWC'06*, pages 544–558, 2006.
10. L. Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, FZI - Research Center for Information Technologies at the University of Karslruhe, 2004.
11. D. Vrandecic, H. S. Pinto, Y. Sure, and C. Tempich. The diligent knowledge processes. *Journal of Knowledge Management*, 9:85–96, 2005.
12. F. Zablith. Dynamic ontology evolution. *ISWC Doctoral Consortium*, 2008.
13. F. Zablith, M. Sabou, M. d'Aquin, and E. Motta. Using background knowledge for ontology evolution. *IWOD*, 2008.