Dynamic Ontology Evolution

Fouad Zablith*

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

Abstract. Ontologies form the core of Semantic Web systems, and as such, they need to evolve to meet the changing needs of the system and its users. Most of current ontology evolution systems require user input during their processes. We propose Evolva, an ontology evolution framework, aiming to substantially reduce or even eliminate user input through exploiting various background knowledge sources. In this paper we present our ontology evolution approach, as well as our preliminary outcomes and future directions.

1 Motivation

Ontology evolution is defined in [1] as the "timely adaptation of an ontology to the arisen changes and the consistent management of these changes". Ontologies form the core of Semantic Web systems, and as such, they need to evolve to meet the changing needs of the system and its users (e.g. new data or functionalities introduced). Evolving an ontology is a time consuming task, and relies on considerable input from a user with knowledge representation skills. A motivating use case is the KMi Semantic Web portal[1] based on the extended AKT reference ontology[2]. The content of this ontology, both in terms of structure and instances, needs to be kept up-to-date by painstaking manual efforts, due, for example, to the continuous introduction of new concepts and instances through the regular publication of KMi news articles.

Our research is motivated by the hypothesis that (online) knowledge and data sources could be explored to significantly decrease or even eliminate user input during the evolution process, thus rendering it dynamic[3]. We understand background knowledge as a range of data sources with various levels of formality: unstructured web pages such as Wikipedia, lexical databases such as WordNet [2] and FrameNet [3], and online ontologies.

* Work funded by the NeOn project under EC grant number IST-FF6-027595

1 http://semanticweb.kmi.open.ac.uk

2 http://kmi.open.ac.uk/projects/akt/ref-onto/index.html

3 In our context, a "dynamic" process involves a degree of automation, which is directly related to the application/data environment in which the ontology evolution occurs. (e.g. the process could be entirely automated in less critical applications, while only a partial automation is feasible for critical ones)

Background knowledge can support finding relationships between new knowledge discovered, and existing knowledge in the domain ontology, which we call "base ontology". For example, WordNet identifies "European Union", a new instance discovered in the KMi news articles, as a type of "Organization", a concept already defined in the base ontology.

Background knowledge helps as well in completing information not available to the scope of the system. Consider the case of adding to the base ontology a publication co-authored by Enrico Motta and Dieter Fensel. KMi's data sources hold significant information about Enrico Motta, who works at KMi. However, Dieter Fensel who is not part of KMi, requires additional effort to identify him as a person with additional details such as current position, address, other publications, etc. These details are, however, available in online Semantic Web data.

2 Research Problem

We focus on evolving ontologies by relying on external background knowledge sources, rather than on user input. Different sub-problems are identified:

Q.1. How to extract new and relevant information? A first step in an ontology evolution process is the identification of new and relevant information that should be added to the base ontology. Such information can reside in text corpora, databases and ontologies, making them a good source of discovering potential changes. These data sources should be processed for identifying and extracting concepts, relations and entities, relevant to the base ontology.

Q.2. How to perform ontological changes dynamically? Extracted information from data sources should be added to the base ontology without relying on user input, with the corresponding relations to existing knowledge. Relations should be discovered automatically by exploring background knowledge sources, and by taking into account the related entities' contextual meaning. For example, "degree" could refer to an *academic degree* in the education context, or to a *unit-of-measure* in the physics context with different types of relations.

Q.3. How to validate the evolved ontology? During evolution, inconsistencies could occur due to conflicting statements, data duplication and temporal related facts. These phenomena are particularly likely to arise when evolution is informed by knowledge extracted from multiple heterogeneous data sources, with various degrees of quality. A temporal inconsistency example is when some of the current KMi news articles mention Peter Scott as KMi's director, conflicting with news published during the period when Enrico Motta was the director of KMi. Such inconsistencies should be identified and resolved.

Q.4. How to manage the evolution? After validating the evolved ontology, its dependent components, such as other ontologies or applications, should be notified with the performed changes for ensuring compatibility. Another requirement is to be able to follow-up the evolution process, and present to the user a degree of control for monitoring and spotting unresolved problems.

3 Related Work

One set of approaches in ontology evolution, such as Klein's [4], focuses on the management of changes performed by users, without exploring external data sources. Another example is Stojanovic [5] who proposed a framework for evolving ontologies mainly triggered by internal sources of change from within the ontology. Klein and Stojanovic encapsulated the ontological changes in an ontology to formulate the types of changes encountered during evolution. Noy et al [6] describe a framework for ontology evolution in collaborative environments. The framework is scenario based and formed of different plugins for Protégé[4]. Additionally, DILIGENT [7] is a collaborative user-centric ontology evolution methodology in which a board of users are responsible for selecting the proposed changes to be applied on the ontology.

Another set of approaches use external data sources for identifying potential ontology changes that are subsequently validated by users. For example DINO [8] is a framework for integrating ontologies. It includes the use of ontology alignment, coupled with agent-negotiation techniques, to generate and select mappings between learned ontologies from text and the base ontology. However, the final integration decision is left to the user. Text2Onto [10] is an ontology learning tool that extracts potential structural and instance changes from text repositories. However it lacks ontology change management and validation. Another example is Dynamo [9], which is a multi-agent system for dynamic ontology construction from domain specific text documents. It is meant to support ontology builders, whose input is a prerequisite to the system.

We note that all described systems rely on user input during the ontology evolution process. Moreover, no system considers the use of background knowledge sources to support evolution.

4 Proposed Approach

We propose Evolva, an ontology evolution framework that explores various background knowledge sources to evolve ontologies and reduce user input. Our framework is visualized in Figure 1 and has the following components:

Information discovery. One way to detect new knowledge to be added to the base ontology is by contrasting it to information contained in external domain and application specific sources, such as text corpora, databases or other ontologies. Text documents contain unstructured data, hence require information extraction or ontology learning tools such as Text2Onto [10]. External ontologies and databases present a more structured source of information, where concepts, relations and instances are explicitly encoded in a well-defined structure. However, a translation should be applied on exploited ontologies to ensure language compatibility with the base ontology. In the case of databases, a transformation should be performed to encapsulate the database schema and entities in an ontology compatible language.

[4] http://protege.stanford.edu

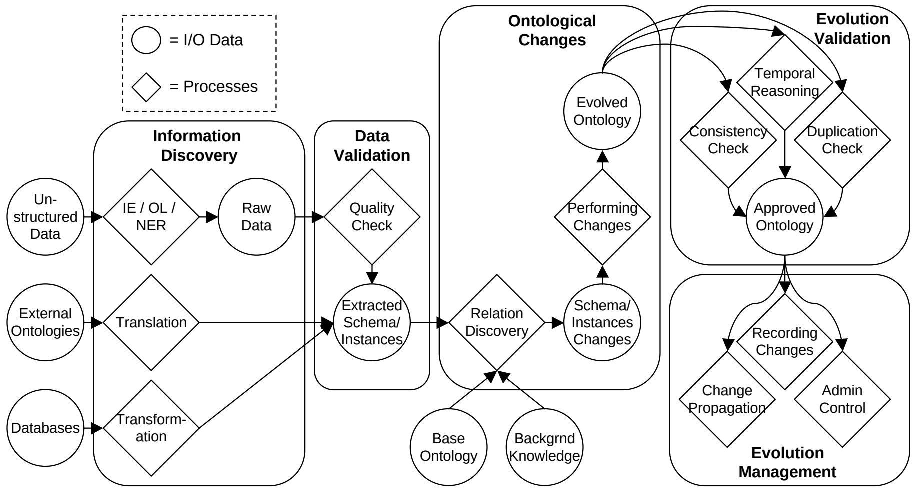

Fig. 1. The Main Components of the Evolva Framework

Data validation. One way to validate discovered information is by applying a set of heuristic rules. For example, most of the two-letter concepts extracted by Text2Onto from KMi's news corpora such as "cu" and "th" are meaningless and should be discarded. Ontologies and databases do not need this kind of low level quality check as the content structure is more trusted.

Ontological changes. The validated schema and instance elements are passed to the the relation discovery process, for resolving the links to existing knowledge. We propose a gradual matching technique, visualized in Figure 2, which starts from the simplest and quickest methods, to the more complex and time-consuming ones: (1) The process starts with a string matching to identify possible equivalence with existing ontology terms. (2) If no equivalence is resolved, i.e. the term is new to the ontology, subsumption relations are attempted to be discovered based on WordNet's senses hierarchy. (3) If no relation is discovered at the level of WordNet, we rely on the Semantic Web for a richer relation discovery process by using the Scarlet relation discovery engine[5], which automatically selects and explores online ontologies [12]. (4) As a final resort, we harvest all the web through search engines APIs, coupled with the use of lexical patterns [13]. In case no relation is found at the final level, the extracted term is discarded, or could be possibly passed for manual check.

We use the discovered relation path for performing the changes to the ontology. A predefined set of encapsulated ontology changes as described in [4] and [5] could be used to perform the changes dynamically, based on the relation path.

Evolution validation. As mentioned earlier, performing ontological changes could generate some problems such as conflicting statements, data duplication and time related inconsistencies. We deal with these problems at the level of

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

the evolution validation component, formed of the consistency and duplication checks, as well as the temporal reasoning process.

Evolution management. The approved ontology is passed to the evolution management component. In this component, the changes performed on the ontology are recorded to ensure functionalities such as tracing or rolling back changes. The changes are then propagated to dependent ontologies and applications. Administrator control is supplied for monitoring purposes, setting the evolution parameters and resolving any additional problem.

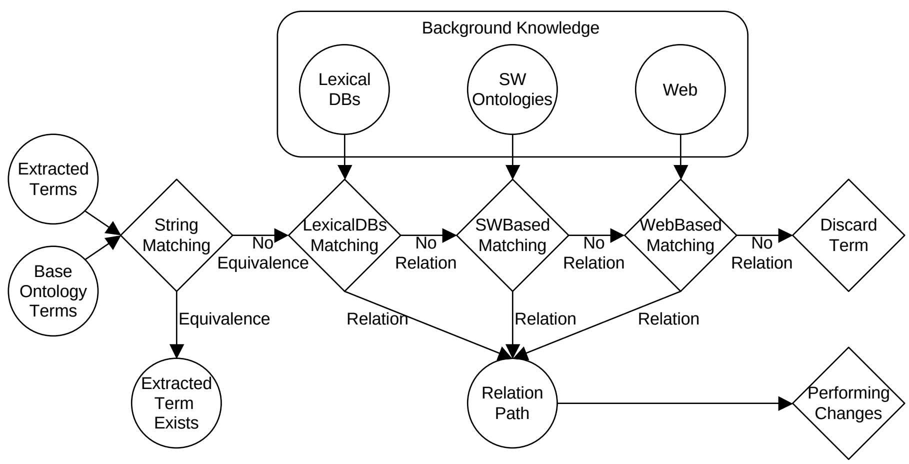

Fig. 2. Gradual Matching Technique

5 Preliminary Outcomes and Future Directions

We are applying our initial methods for information discovery and data validation on the KMi portal data to evolve the underlying AKT ontology. We use Text2Onto [10] to extract information from the KMi Planet News web-pages[6]. We then perform string matching between the extracted concepts and the current KMi ontology, based on the Jaro-Winkler [14] algorithm. We are currently testing the potential use of WordNet [2] for relation discovery. The Wu and Palmer similarity [15] is used. The results seem promising so far, such as finding relationship paths between "Presentation" and "Activity" and between "European Union" and "Organization".

As our proposed framework spans over various branches of research, it will be difficult to highly contribute in all of them. We will focus our contributions on the ontological changes and evolution validation. After finalizing the WordNet

[6] http://news.kmi.open.ac.uk

experiment, we plan to proceed with the relation discovery tests. This task is scheduled to be finalized by the end of the first year of this PhD (October 2008). Then we tackle the implementation of performing ontological changes dynamically, and finally focus on the evolution validation and management. After the implementation phase, we plan to test Evolva in other environments such as the UN's Food and Agriculture Organization (FAO)[7] domain. We will then work on setting the basis of evaluation procedures, for example by testing the accuracy and degree of coverage of our system by comparing it to an ontology engineer's performance. This will enable further extensions and improvements to Evolva.

References

1. Haase, P. and Stojanovic, L.: Consistent Evolution of OWL Ontologies. Proc. of ESWC. (2005) 182–197
2. Fellbaum, C.: Wordnet: An Electronic Lexical Database. MIT Press (1998)
3. Baker, C. F., Fillmore, C.J. and Lowe, J.B.: The Berkeley FrameNet project. Proc. of the COLING-ACL. **1** (1998) 86–90
4. Klein, M.: Change Management for Distributed Ontologies. PhD thesis, Vrije Universiteit Amsterdam. (2004)
5. Stojanovic, L.: Methods and Tools for Ontology Evolution. PhD thesis, University of Karlsruhe. (2004)
6. Noy, N., Chugh, A., Liu, W. and Musen, M.: A Framework for Ontology Evolution in Collaborative Environments. Proc. of ISWC. (2006) 544–558
7. Vrandečić, D., Pinto, H. S., Sure, Y. and Tempich, C.: The DILIGENT Knowledge Processes. Journal of Knowledge Management. **9**(5) (2005) 85–96
8. Nováček, V., Laera, L. and Handschuh, S.: Semi-automatic Integration of Learned Ontologies into a Collaborative Framework. Proc. of IWOD. (2007)
9. Ottens, K., Gleizes, M. P. and Glize, P.: A Multi-Agent System for Building Dynamic Ontologies. Proc. of AAMAS. (2007)
10. Cimiano, P. and Völker, J.: Text2Onto A Framework for Ontology Learning and Data-driven Change Discovery. Proc. of NLDB. (2005) 227–238
11. d'Aquin, M., Baldassarre, C., Gridinoc, L., Sabou, M. Angeletou, S. and Motta, E.: Watson: Supporting Next Generation Semantic Web Applications. Proc. of WWW/Internet conference. (2007)
12. Sabou, M., d'Aquin, M. and Motta, E.: Exploring the Semantic Web as Background Knowledge for Ontology Matching. Journal of Data Semantics. (2008)
13. Cimiano, P., Handschuh, S. and Staab, S.: Towards the Self-Annotating Web. Proc. of WWW. (2004) 462–471
14. Cohen, W. W., Ravikumar, P. and Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. Proc. of the IJCAI Workshop on Information Integration on the Web. (2003)
15. Wu, Z. and Palmer, M.: Verb Semantics and Lexical Selection. Proc. of the 32nd Annual Meeting of the ACL. (1994) 133–138

[7] http://www.fao.org