

# Ontology Evolution with Evolva

Fouad Zablith, Marta Sabou, Mathieu d'Aquin, and Enrico Motta

Knowledge Media Institute (KMi), The Open University  
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom  
{f.zablith,r.m.sabou,m.daquin,e.motta}@open.ac.uk

**Abstract.** Ontology evolution is a painstaking and time-consuming process, especially in information rich and dynamic domains. While ontology evolution refers both to the adaptation of ontologies (e.g., through additions or updates possibly discovered from external data sources) and the management of these changes, no existing tools offer both functionalities. The Evolva framework aims to be a blueprint for a comprehensive ontology evolution tool that would cover both tasks. Additionally, Evolva proposes the use of background knowledge sources to reduce user involvement in the ontology adaptation step. This demo focuses on the initial, concrete implementation of our framework.

## 1 Introduction

Ontologies form the backbone of Semantic Web systems. In today's information rich environments, ontologies need to be kept up-to-date so that the dependent systems remain operational. We highlight two research approaches to this problem: The first considers ontology evolution as the management of user-performed changes [5,7,9,10], while the second primarily focuses on dynamically updating and learning ontologies [1,2,6]. We understand ontology evolution as the “timely adaptation of an ontology to the arisen changes and the consistent management of these changes” [4]. This definition reflects the need of combining the two aforementioned approaches for supporting both evolution tasks. Yet, to our knowledge, no practical and complete solutions exist that cover all stages of ontology evolution.

Our ontology evolution framework, Evolva, proposes a comprehensive approach to evolution by identifying tasks that need to be performed both during ontology adaptation and change management (Section 2). The Evolva tool (Section 3 and 4) is a partial, concrete implementation of the framework. One of its main novelties, which will be captured by the demo, is in pioneering the use of background knowledge to decrease user input [12].

## 2 Underlying Approach: The Evolva Framework

The Evolva framework (detailed in [11]) consists of five main components as shown in Figure 1 and described next.

*Information Discovery.* New information can be potentially discovered in external data sources. Contrasting the ontology with the content of such sources is

a way of detecting new knowledge that should be reflected by the base ontology. Data sources can be either unstructured (e.g., text documents, folksonomy tagspaces) or structured (e.g., databases, ontologies). This component handles each data source differently: (1) Text documents are processed using information extraction, ontology learning or entity recognition techniques. (2) Other external ontologies are subject to translation for language compatibility with the base ontology, and (3) database content is translated into ontology languages.

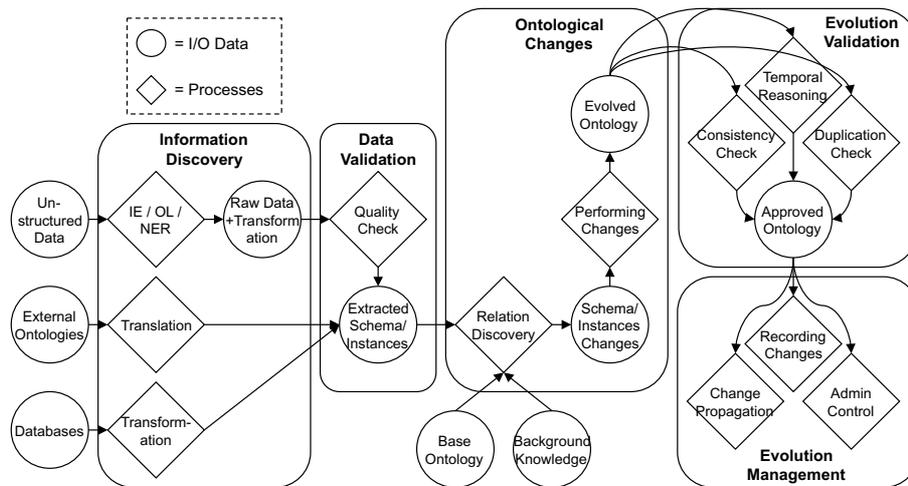


Fig. 1. Evolve Framework Overview

*Data Validation.* Discovered information is validated in this component. We rely on a set of heuristic rules such as the length of the extracted terms. This is especially needed for information discovered from text documents, as information extraction techniques are likely to introduce noise. For example, most of the less than two-letter terms extracted from a corpora are meaningless and should be discarded. In the case of structured data, this validation is less needed as the type of information is explicitly defined.

*Ontological Changes.* This component is in charge of establishing relations between the extracted terms and the concepts in the base ontology. These relations are identified by exploring a variety of background knowledge sources. Appropriate changes are then performed on the ontology.

*Evolution Validation.* 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 the evolution validation component, formed of the consistency and duplication checks, in addition to temporal reasoning.

*Evolution Management.* Managing the evolution is about recording changes, and giving the ontology curator a degree of control over the evolution, as well as

propagating changes to other dependent ontologies (e.g., in the case of an ontology network) or applications. User control will deal with tracking ontology changes, spotting and solving unresolved problems.

### 3 Implementation: The Evolva NeOn Toolkit Plugin

The Evolva NeOn toolkit plugin is a concrete implementation of our framework and will be the object of the proposed demonstrations (see screenshot in Figure 2). Currently, the plugin supports the first three steps depicted in Figure 1.

In the *information discovery* step Text2Onto is used to identify concepts in a corpus of text documents (selected in the “Data Sources” panel). *Data validation* relies on string similarity and term length measures. We use a customizable Jaro-based [3] calculation to compute the similarity of the extracted and existing concepts. Concepts with the similarity under a given threshold are considered new and are validated based on their length. The user can set the parameters of the validation techniques and also manually indicate which concepts should be considered for integration (“Data Validation”).

*Ontological changes* are identified by finding links between validated terms and ontology concepts. We use two main sources of background knowledge: WordNet and online ontologies accessed through Scarlet<sup>1</sup>. Scarlet [8] is a Semantic Web based relation discovery engine. It automatically selects and explores online ontologies to discover relations between two given concepts. For example, when relating two concepts labeled *Researcher* and *AcademicStaff*, Scarlet 1) identifies (at run-time) online ontologies that can provide information about how these two concepts inter-relate and then 2) combines this information to infer their relation. [8] describes two increasingly sophisticated strategies to identify and to exploit online ontologies for relation discovery. Hereby, we rely on the first strategy that derives a relation between two concepts if this relation is defined (directly or through transitivity) within a single online ontology. Besides subsumption relations, Scarlet is also able to identify disjoint and named relations. As in the case of WordNet, the derived relations are accompanied by a path of inferences that lead to them (“Relation Discovery” panel). We performed an experiment on the potential use of background knowledge in our context, and we got a promising 77% precision<sup>2</sup>. A list of ontology changes are deduced from the relation discovery step and applied to the ontology (“Ontology Changes”).

### 4 Demonstration Plan

The demonstration will focus on the various functionalities of the Evolva plugin (Figure 2) that support ontology evolution from textual documents. A particular emphasis will be on the use of background knowledge sources to automatically identify relations between terms and ontology concepts. This gives our tool a

<sup>1</sup> <http://scarlet.open.ac.uk>

<sup>2</sup> Additional details of the experiment can be found in [12]

cutting edge advantage in decreasing user input. It is a critical feature as this is where user input is usually most needed. The following functionalities will be demonstrated:

**Ontology selection and loading in NeOn toolkit.** At the left side of Figure 2, the user will see how to load and open an ontology in the NeOn toolkit, and start Evolva.

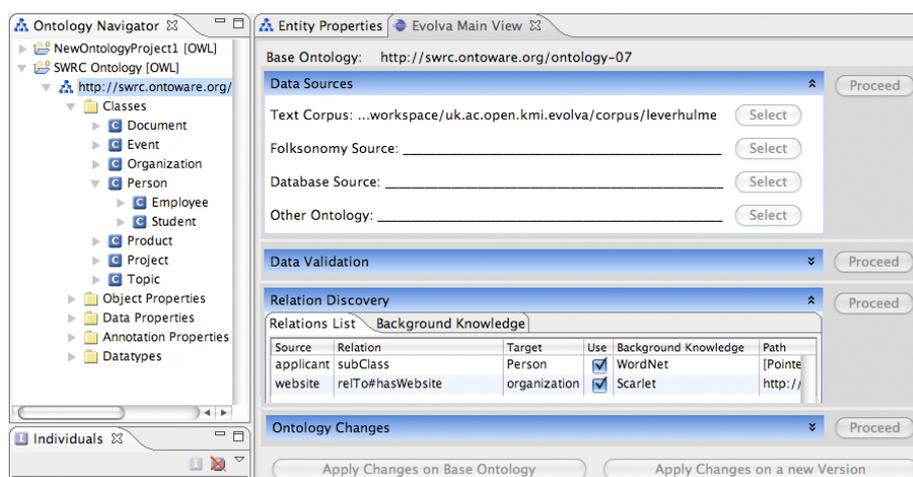


Fig. 2. Evolva Pilot System Screenshot

**The selection of data sources.** The user selects the text documents from where terms are extracted, then proceeds to the next step in the process.

**Data validation and parameters settings.** At this level the list of extracted terms are displayed to the user. The validation shows the terms that correspond to already existing ontology concepts. Further noise is removed based on the term length. This is performed by a customizable string length checker to exclude for example one or two letter terms. Parameters for controlling the automated validation include changing the minimum length of acceptable terms and controlling the similarity threshold. In addition to the automatic validation, the user is able to manually select which terms should be included in the following step.

**Relation discovery between terms and ontology concepts.** The user will experience how background knowledge helps identifying links between the validated terms and ontology concepts. For example, consider the case of the base ontology containing the concepts *Person* and *Organization*. From the text documents, Evolva discovers *Applicant* and *Website* as potential concepts to be added to the ontology. WordNet is able to link *Applicant* as *subClassOf Person*, and online ontologies link the new concept *Website* through a *hasWebsite* relation to *Organization*. The “Relation Discovery” panel shows these relations and allows their inspection. The complete relation path is displayed to help in verifying the

relation's correctness. This panel also contains a set of customizable parameters such as: the background knowledge source to be used (WordNet, Scarlet or both); the maximum inference path for deriving a relation; the type of relations to be discovered.

**Applying changes to the ontology.** Based on the discovered relations, the user will be able to see the list of changes to be applied to the ontology. This includes the type of changes and their corresponding details. At the bottom of the screen, the user has the choice to apply the changes either on the base ontology or to create a new detached version. In the latter case, the new ontology version will appear in the left panel, having the current date and time appended to its URI.

## Acknowledgements

This work was funded by the NeOn project sponsored under EC grant number IST-FF6-027595.

## References

1. Alani, H., Harris, S., O'Neil, B.: Winnowing ontologies based on application use. In: Sure, Y., Domingue, J. (eds.) *ESWC 2006*. LNCS, vol. 4011, pp. 185–199. Springer, Heidelberg (2006)
2. Bloehdorn, S., Haase, P., Sure, Y., Voelker, J.: *Ontology Evolution*, pp. 51–70. John Wiley & Sons, Chichester (2006)
3. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: *Proceedings of the IJCAI 2003 Workshop on Information Integration on the Web (IIWeb 2003)* (2003)
4. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) *ESWC 2005*. LNCS, vol. 3532, pp. 182–197. Springer, Heidelberg (2005)
5. Klein, M.: *Change Management for Distributed Ontologies*. PhD thesis, Vrije Universiteit in Amsterdam (2004)
6. Novacek, V., Laera, L., Handschuh, S.: Semi-automatic integration of learned ontologies into a collaborative framework. In: *IWOD* (2007)
7. Noy, N.F., Chugh, A., Liu, W., Musen, M.A.: A framework for ontology evolution in collaborative environments. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006*. LNCS, vol. 4273, pp. 544–558. Springer, Heidelberg (2006)
8. Sabou, M., d'Aquin, M., Motta, E.: Exploring the Semantic Web as Background Knowledge for Ontology Matching. *Journal on Data Semantics XI* (2008)
9. Stojanovic, L.: *Methods and Tools for Ontology Evolution*. PhD thesis, FZI (2004)
10. Vrandecic, D., Pinto, H.S., Sure, Y., Tempich, C.: The DILIGENT knowledge processes. *Journal of Knowledge Management* 9, 85–96 (2005)
11. Zablith, F.: *Dynamic ontology evolution*. ISWC Doctoral Consortium (2008)
12. Zablith, F., Sabou, M., d'Aquin, M., Motta, E.: Using background knowledge for ontology evolution. In: *IWOD Workshop at ISWC* (2008)