Ontology Learning for Semantic Web Services

Dr. Paul Buitelaar, DFKI GmbH, Saarbrücken, Germany

We describe the central role of ontologies in knowledge markup for semantic web services. Markup of web services with ontology-based metadata will allow for dynamic discovery and composition. However, because ontologies evolve over time and between applications, methods and tools will need to be developed for automatic learning and adaptation of ontologies. Here we will focus on methods for ontology learning from text (i.e. from document collections corresponding to the application domain of the ontology under consideration).

1 Introduction

In recent years, the Internet evolved from a global medium for information exchange (directed mainly towards human users) into a "global, virtual work environment" (for both human users and machines). Building on the world-wide-web, developments such as G*rid technology*, *web services* and the S*emantic Web* contributed to this transformation, the implications of which are now slowly but clearly being integrated into all areas of the new digital society (e-business, e-government, e-science, etc.) In particular, Grid technology allows for distributed computing (Foster and Kesselman, 1999), web services for a distributed workflow (e.g. by use of WSDL and WSFL), and the Semantic Web for increasingly intelligent and therefore autonomous processing (Berners-Lee et al., 2001).

As illustrated in Figure 1., semantic web services are based on an infrastructure of ontology-based knowledge markup of resources and services, which ensures that a common understanding will exist between machine processes, as well as between processes and human users.

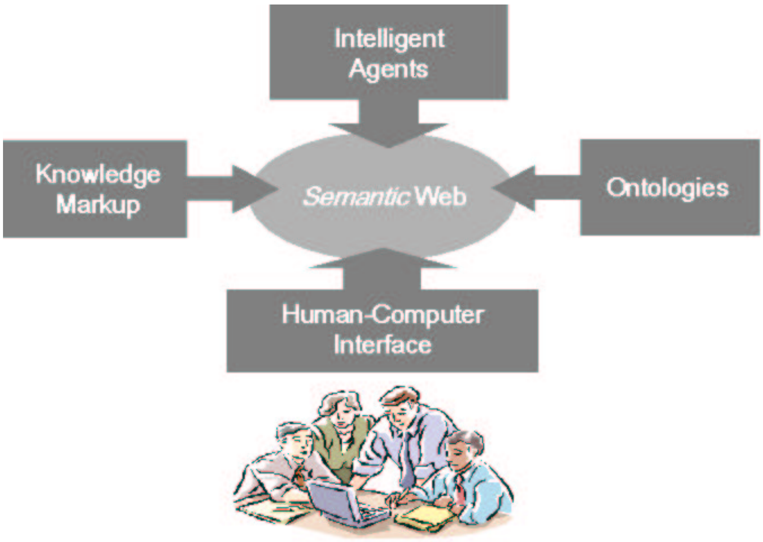


Figure 1: The Semantic Web Vision

2 Semantic Web Services

The Semantic Web enables dynamic discovery of web services through standardized markup with semantic descriptions of the functionality of a given web service. Instead of working only with fixed web service compositions, the Semantic Web allows for the submission of a semantic description of the required functionality of a web service (e.g. to a web service registry), in order to 'discover' any number of suitable (semantic) web services that will be able to accomplish a required task.

Knowledge markup for semantic web services is based on a common (shared) understanding of a given domain and the concepts and tasks relevant to this domain. The shared knowledge will be represented in an ontology, which formally defines objects and their properties (classes, relations) of the domain. The definition of such an ontology may be done manually, but can be supported also by semi-automatic tools that allow for the extraction of relevant concepts from related databases and document collections.

3 Ontology-based Knowledge Markup

Knowledge markup is an elaboration of metadata as currently used for a restricted set of applications, e.g. the Dublin Core set of bibliographical metadata such as 'title, 'author', etc. It is to be expected that within the next years ontologies for many more such applications will be formally encoded and commonly used. Specifically in the context of e-business this will become apparent, as integrated sections of industry will need a common and explicit understanding of their products and business processes in order to allow for an increasingly automatic commercial exchange.

The definition of knowledge representation languages for ontology definition and for ontology-based knowledge markup is currently an active field of study, which has led to a number of emerging standards. Foremost among these are RDF-Schema and OWL, besides XML-Schema and TopicMaps.

From a semantic point of view, ontologies defined in XML-Schema, RDF-Schema or OWL are all based on the notion of a namespace, which defines the interpretation context of any XML-S, RDF-S or OWL expression. For instance, the following XML markup ensures that the job of *John Smith* as a **systems-analyst** is interpreted exactly as defined in the ontology at the indicated namespace.

<xmnls:jobs="http://www.jobs.org/owl-jobs-ontology#">
<jobs:systems-analyst>*John Smith***</jobs:systems-analyst>**

In this way, a semantic web application will be able to identify *John Smith* as a **systems-analyst** and look up additional knowledge on this concept in the **owl-jobs-ontology**, which it can access at the indicated namespace address. The ontology would represent for instance the knowledge that the class **systems-analyst** has property **salary** with a value of class **salary-scale**, which in turn will be defined formally in the **owl-jobs-ontology**. In this way, a semantic web banking service would be able to identify John Smith as a **systems-analyst,** infer the corresponding **salary-scale** from the **owl-jobs-ontology,** automatically match this onto a range of available banking services and contact John Smith on a corresponding offer.

4 Ontology Learning

Ontologies are domain descriptions that tend to evolve rapidly over time and between different applications (see e.g. Noy and Klein, 2002). Currently however, ontologies are often developed with a specific goal in mind, without much consideration for this dynamic aspect of ontology development. Still, it is ineffective and costly to build ontologies for a new purpose each time from scratch, which may cause a major barrier to their large-scale use in knowledge markup for semantic web services. For this reason there have been recent developments in learning or adapting ontologies dynamically, e.g. by analysis of a corresponding knowledge base (Deitel et al., 2001, Suryanto and Compton, 2001) or document collection (i.e. ontology learning from text), which is also a clear option as human language is a primary mode of knowledge transfer.

4.1 Ontology Learning from Text

A number of systems have been proposed for ontology learning from text, e.g.: ASIUM (Faure et al., 1998), TextToOnto (Maedche and Staab, 2000), Ontolearn (Navigli et al., 2003). Most of these depend on linguistic analysis and machine learning algorithms to find potentially interesting concepts and relations between them.

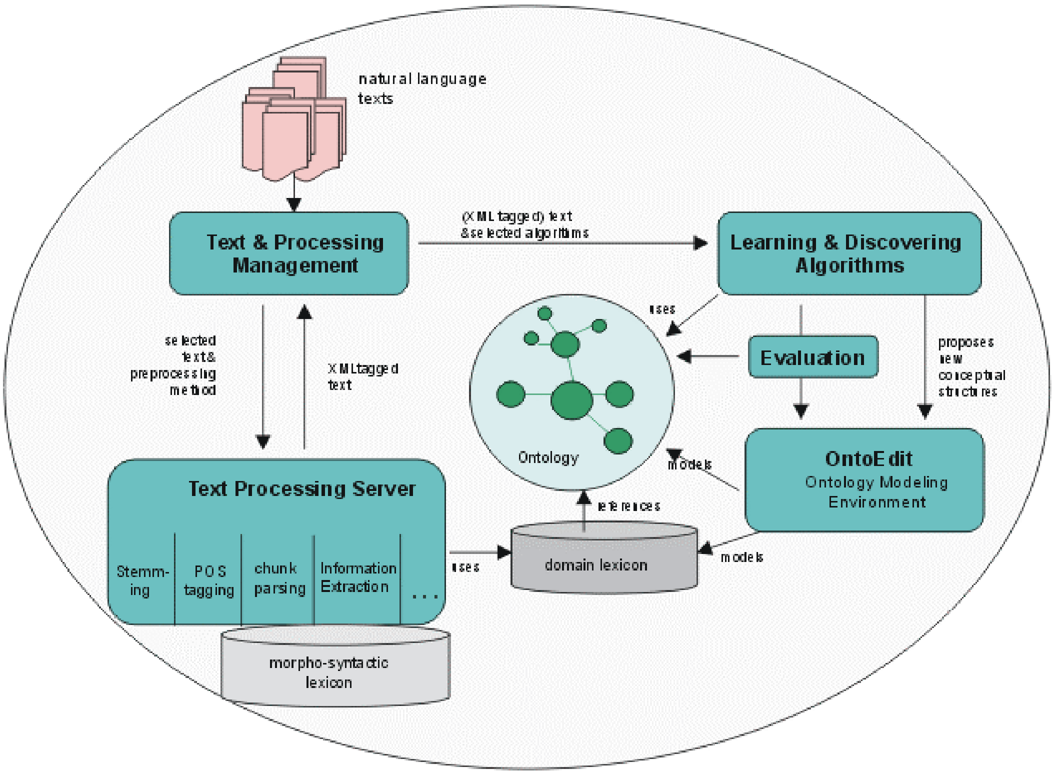


Figure 2: The TextToOnto Approach

An overview of the TextToOnto system is shown in Figure2. The basic approach is:

- to collect a set of documents relevant to the application area of the ontology we would like to develop with the system
- to analyze and annotate these documents linguistically by use of language technology tools
- to extract a database of occurring terms from the annotated document collection
- to identify possible relations between these terms by use of machine learning algorithms (i.e. by use of the „association rules“ algorithm, which identifies correlations in the co-occurrence of classes of objects in a data collection)
- to represent the identified terms and relations as classes with assigned properties in a formal ontology

In this way, the TextToOnto approach allows for semi-automatic ontology development, in which the defined ontology is based on and has a clear relation to relevant documents within an organisation. The advantage of such an approach is a feedback option between ontology construction and use on the one hand (formal information and knowledge flow – implemented business processes) and document production and use (informal information and knowledge flow – email, memos, etc.) on the other. The evolving knowledge structure within an organisation may therefore be monitored through ongoing linguistic analysis of incoming and produced documents, the result of which will be used in updating the corresponding ontology.

4.2 OntoLT

The TextToOnto approach relies on a certain level of linguistic analysis for the extraction of possible classes and properties. Here, we introduce the OntoLT approach (Buitelaar et al., 2003), which is currently under development at DFKI. OntoLT relies even more on linguistic knowledge through its use of built-in patterns that map possibly complex linguistic structure directly to concepts and relations. OntoLT will provide a complete integration of ontology extraction from text into the widely used Protégé ontology development environment (see e.g. Knublauch, 2003), which allows for efficient handling and exchange of extracted ontologies.

The approach provides a plug-in for Protégé, with which concepts (Protégé classes) and relations (Protégé slots) can be extracted automatically from annotated text collections. For this purpose, the plug-in defines a number of linguistic patterns over an annotation format that will automatically extract class and slot candidates. Alternatively, the user can define additional rules, either manually or by the integration of a machine learning process.

The annotation format that is used by the OntoLT system integrates multiple levels of linguistic annotation into separate tracks with options of reference between them via indices (based on Buitelaar and Declerck, 2003). Linguistic annotation covers part-of-speech tagging; morphological analysis (stemming and decomposition of complex words such as *Flachbildschirm* in German, which may be decomposed into *Flach*, *Bild* and *Schirm*); semantic tagging (identification of concepts and their synonyms); and semantic parsing (identification of relations between concepts by analysis of occurring verbs and their linguistic subjects and objects).

Consider a small example in the development of an ontology for the computer science field from a collection of relevant documents (i.e. scientific papers). OntoLT will allow for the automatic extraction and representation of classes of technology (e.g., “web services”, “P2P platforms”, “RDF parsing”) from this document collection. In fact, this knowledge can be extracted from such sentences as:

> *… the Database Group at Stanford University develops an open source P2P platform…*

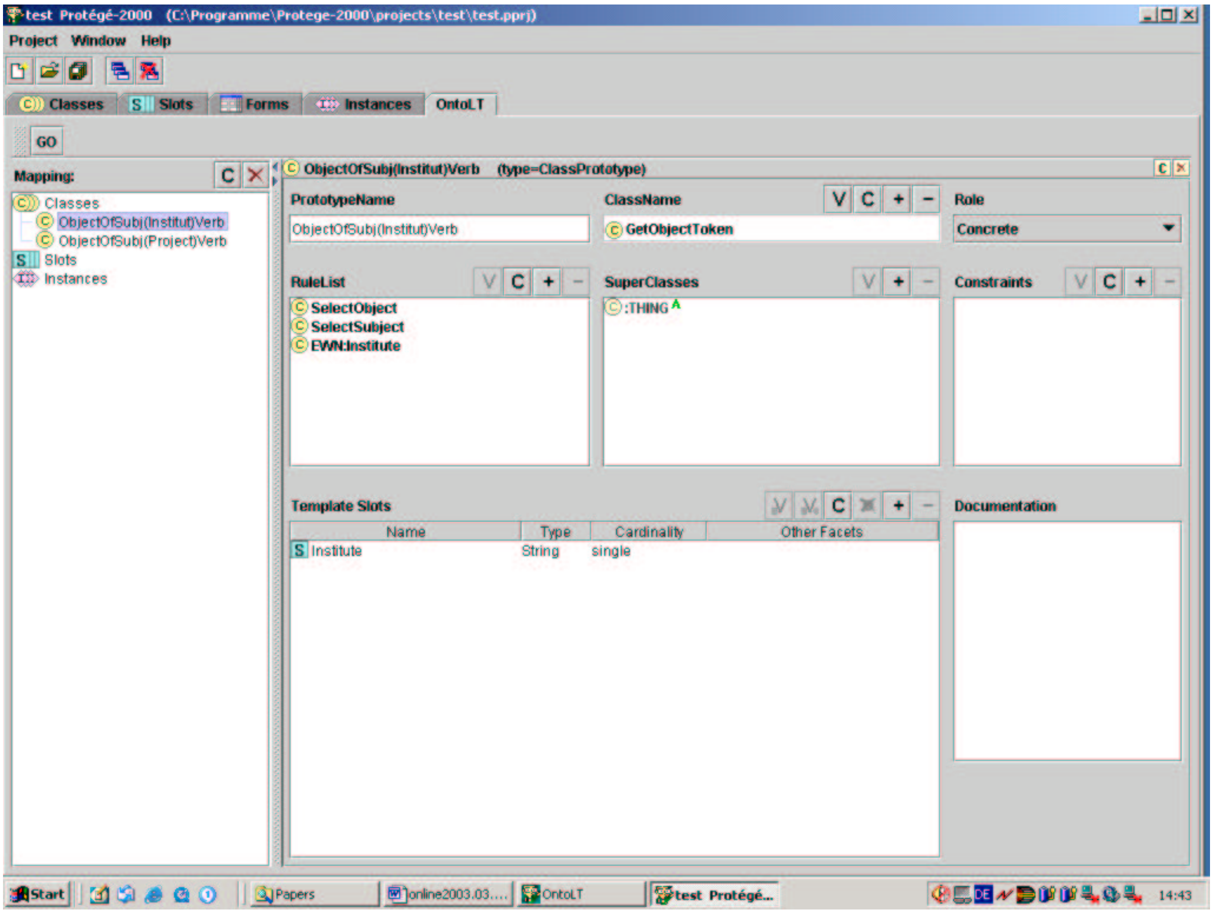
Linguistic analysis and annotation produces the following structured representation of identified terms and relation(s) in this sentence:

```
<relations>
        </relation id="r1" name="Develop" verb="w7" subj="t1" obj ="t2"/>
</relations>

<terms>
        </term id="t1" from="w1" to="w6" type="NP" sem_class ="Institute"/>
        </term id="t2" from="w8" to="w12" type="NP"/>
</terms>

<text>
        <token id="w1" pos="DT" lemma="the">the</token>
        <token id="w2" pos="NN" lemma="database">Database</token>
        <token id="w3" pos="NN" lemma="group">Group</token>
        <token id="w4" pos="IN" lemma="at">at</token>
        <token id="w5" pos="NNP" lemma="Stanford">Stanford</token>
        <token id="w6" pos="NNP" lemma="university">University</token>
        <token id="w7" pos="VBZ" lemma="develop">develops</token>
        <token id="w8" pos="DT" lemma="a">an</token>
        <token id="w9" pos="JJ" lemma="open">open</token>
        <token id="w10" pos="NN" lemma="source">source</token>
        <token id="w11" pos="NN">P2P</token>
        <token id="w12" pos="NN" lemma="platform">platform</token>
</text>
```

Documents annotated in this way can then be imported in the OntoLT system, which allows for automatic mapping of identified terms and relation(s) to Protégé classes and/or properties as shown in Figure 3. below:


test Protégé-2000 (C:\Programme\Protege-2000\projects\test\test.pprj)
Project
Window
Help
Classes
Slots
Forms
Instances
OntoLT
GO
Mapping:
Classes
ObjectOfSubj(Institut)Verb
ObjectOfSubj(Project)Verb
Slots
Instances
ObjectOfSubj(Institut)Verb (type=ClassPrototype)
Documentation
Start
Papers
online2003.03....
OntoLT
test Protégé...
14:43

PrototypeName: ObjectOfSubj(Institut)Verb

ClassName: GetObjectToken

Role: Concrete

RuleList: SelectObject, SelectSubject, EWN:Institute

SuperClasses: :THING

Constraints:

Template Slots

Name	Type	Cardinality	Other Facets
Institute	String	single	

Figure 3: The OntoLT Plug-in for Protégé

For instance, by selecting the **Institute-Verb-Obj** pattern, the system selects all terms of semantic class **Institute** (i.e., *-the Database Group at Stanford- University*) that are the linguistic subject of any verb that expresses a certain relation (e.g. *develop, design, implement* for the relation **Develop**). Subsequently, the user is presented with a list of automatically generated Protégé classes corresponding to the extracted linguistic objects of these verbs (i.e. *-an open source- P2P platform*). In this way, OntoLT will automatically execute all selected patterns and interactively construct a formal ontology on the basis of the extracted terms and relations. The user may be prompted on the preferred sequence in execution of the selected patterns or a default sequence may be applied.

5 Conclusions

Ontology learning tools will be an important component of Semantic Web applications, for instance to allow for automatic updates of the knowledge markup of semantic web services. As business processes change, corresponding ontologies need to evolve, as well as the semantic web services descriptions that depend on them. Ontology learning from corresponding document collections could play a growing role in this process.

References

- Berners-Lee, T., Hendler, J. and Lassila O. *The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American. May, 2001.

- Buitelaar P. and Declerck Th. *Linguistic Annotation for the Semantic Web* In: Siegfried Handschuh, Steffen Staab (eds.) *Annotation for the Semantic Web*, IOS Press, January, 2003

- Buitelaar P., Olejnik D. and Sintek M. *OntoLT: A Protégé Plug-In for Ontology Extraction from Text* Submitted to the Demo Session of the International Semantic Web Conference (ISWC) 2003, Sunibel Island, Florida, October 2003.

- Deitel A., Faron C. and Dieng R. *Learning Ontologies from RDF Annotations* In: Proceedings of the IJCAI Workshop on Ontology Learning, Seattle, Washington, 2001.

- Faure D., Nédellec C. and Rouveirol C. *Acquisition of Semantic Knowledge using Machine Learning Methods: The System ASIUM* Technical Report ICS-TR-88-16, 1998.

- Foster I. And Kesselman C. (eds.) *The Grid: Blueprint for a New Computing Infrastructure*. Morgan-Kaufmann, 1999.

- Knublauch H. *An AI Tool for the Real World: Knowledge Modeling with Protégé* JavaWorld, June 20, 2003.

- Maedche, A., Staab, S.: *Semi-automatic Engineering of Ontologies from Text.* In: Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, 2000.

- Navigli R., Velardi P., Gangemi A. *Ontology Learning and its application to automated terminology translation* IEEE Intelligent Systems, vol. 18:1, January/February 2003.

- Noy N. F. and Klein M. *Ontology Evolution: Not the Same as Schema Evolution* In: *Knowledge and Information Systems*, in press. Available as technical report SMI-2002-0926, 2002.

- Suryanto H. and Compton P. *Discovery of Ontologies from Knowledge Bases* In: Proceedings of the First International Conference on Knowledge Capture, Victoria, BC, Canada, October 2001.

- Dublin Core: http://dublincore.org/

- OWL: http://www.w3.org/TR/owl-features/

- RDF Schema: http://www.w3.org/TR/rdf-schema/

- TopicMaps: http://www.topicmaps.org/xtm/1.0/

- XML Schema: http://www.w3.org/XML/Schema

- WSDL: http://www.w3.org/TR/wsdl

- WSFL: http://www.ebpml.org/wsfl.htm