Semantic Versioning Manager: Integrating SemVersion in Protégé

Tudor Groza
DERI, National University of Ireland, Galway, Ireland
www.deri.org
tudor.groza@deri.org

Max Völkel
FZI, Forschungszentrum Informatik, Karlsruhe, Germany
www.fzi.de
voelkel@fzi.de

Siegfried Handschuh
DERI, National University of Ireland, Galway, Ireland
www.deri.org
siegfried.handschuh@deri.org

ABSTRACT

Knowledge domains and their semantic representations via ontologies are typically subject to change in practical applications. Additionally, engineering of ontologies often takes place in distributed settings where multiple independent users interact. Therefore, change management for ontologies becomes a crucial aspect for any kind of ontology management environment. We introduce a new RDF-centric versioning approach and an implementation called SemVersion integrated as the Semantic Versioning Manager plug-in in Protégé. SemVersion provides structural and semantic versioning for RDF models and RDF-based ontology languages like RDFS.

INTRODUCTION

For many practical applications, ontologies (cf. Staab and Studer 2004) can not be seen as static entities, they rather change over time. Support for change management is crucial to support uncontrolled, decentralized and distributed engineering of ontologies. First approaches have been described in (Klein 2004 and Stojanovic 2004). But, there is no tool that functions as a standard versioning system for ontologies like CVS does in the field of software development.

We introduce an RDF-based approach that provides versioning for RDF models and RDF-based ontology languages like RDFS, OWL flavors or TRIPLE (Sintek and Decker 2002). We present a working methodology accompanied by its implementation in the system SemVersion. We then integrate it in Protégé as a tab plug-in, the Semantic Versioning Manager Tab, in order to provide to the end-users a natural way of editing and versioning their ontologies.

Our approach is inspired by the classical CVS system for version management of textual documents (e.g. Java code). The core element of our approach is the separation of language-specific features (the semantic diff) from general features, such as structural diff, branch and merge, management of projects and metadata.

A first survey on causes and consequences of changes in an ontology is presented in (Klein and Fensel 2001), followed by OntoView, an implementation for ontology versioning (Klein and Fensel 2002) that is based on the comparison of two ontology versions in order to detect changes. Basically, the system compares ontological classes, displays them side-by-side in RDF/XML and leaves it to the user to state "identical" or "conceptual change".

ISOCO KPOntology[1] is another example of a library, somehow similar to our system. It provides a high level API for managing ontologies, with support for multiple different triple stores. While KPOntology focuses on the ontology management, SemVersion's primary focus is on versioning, the management aspects being transparent to the end user.

SEMANTIC VERSIONING MANAGER TAB

The Semantic Versioning Manager Tab (SVM Tab) is our solution for integrating SemVersion in the most natural environment for creating and editing ontologies, i.e. Protégé. The Ontology Lifecycle has 4 phases:

[1] http://kpontology.isoco.com/

Creation/generation, versioning, evaluation/visualization and negotiation. The goal of this integration is to add a plus of functionality in order to provide to the end-user within Protégé, the package of functions mapping to three of the four phases of the ontology lifecycle.

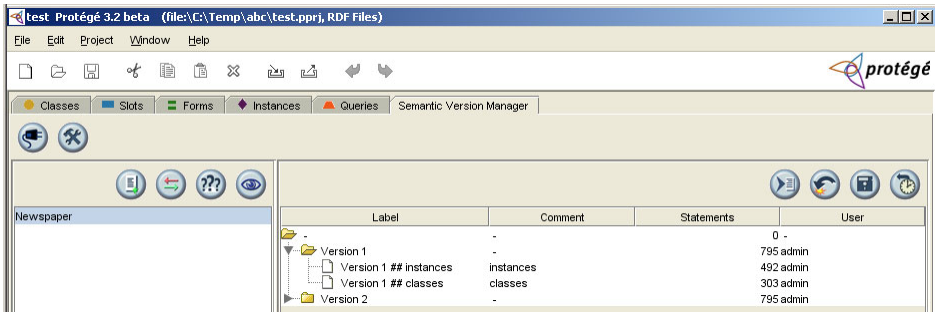

Fig. 1. The Semantic Versioning Manager Tab in Protégé

As core functionalities, the versioning manager tab offers the possibility to:

- create new versioned models (an ontology under version control)
- add versions to a particular versioned model from:
  - the current edited ontology
  - external sources – now it accepts only local files, but in the future, it will accept also URIs
- export a particular version as a file (RDF syntax)
- load a particular version as the current editable knowledge base

Because SemVersion is an RDF-based versioning system, the provided diff operations have to be considered from the RDF semantics point of view. The structural diff represents the set-theoretic difference of two RDF triple sets. Following, we will describe how the semantic diff is achieved: consider two models A and B that are versions of the same RDF Schema model. In order to compute the semantic diff, we use RDF Schema entailment on model A and infer all triples we can (*Inf*(A)). We apply the same operation for model B (*Inf*(B)). Then we calculate a structural diff on *Inf*(A) and *Inf*(B). To summarize, a way to compute a semantic diff is thus to materialize the complete entailment (transitive closure) of two RDFS models and then perform a structural diff on the transitive closure.

The current visualizations provided for the structural and semantic diffs are in terms of statements. Our goal is to offer a more intuitive visualization, for example, by displaying the two ontologies in parallel and create graphic connections to indicate the added and removed statements. Note that the Protégé Prompt Tab has a very clear presentation of the structural diff and represents a good example for us. We also intend to build a graphical visualization for the semantic diff, since the information provided by it, is more expressive and intuitive.

The Semantic Versioning Manager Tab has a graphical visualization for the structural diff, but only in terms of classes and subclasses between several versions. This functionality was created using the Aduna Cluster Map Library[2] and it is depicted in Fig. 2. The idea behind the visualization is to emphasize the differences between several versions by creating clusters of common subclasses of the same class in different versions of the same ontology. By checking more entities, the graph will grow, displaying all the selected classes together with their relationships and clusters of subclasses (an arrow in the graph represents a parent-child relationship).

In terms of implementation, we wanted as a first step to reuse the functionalities provided by Protégé for dealing with RDF ontologies. That is why, two of the most important functions of the SVM Tab (create version from current ontology and load version as current knowledge base) are realized by using the Protégé RDF Backend. Although in terms of reutilization, this was a good design decision, it also introduced a constraint which has to be taken into account by the end-user: the Protégé project on which the user is working has to have an RDF knowledge base. This limitation will be corrected as soon as we will implement our own stream-based RDF importing mechanism, part of the second step of development.

[2] http://aduna.biz/products/technology/clustermap/index.html

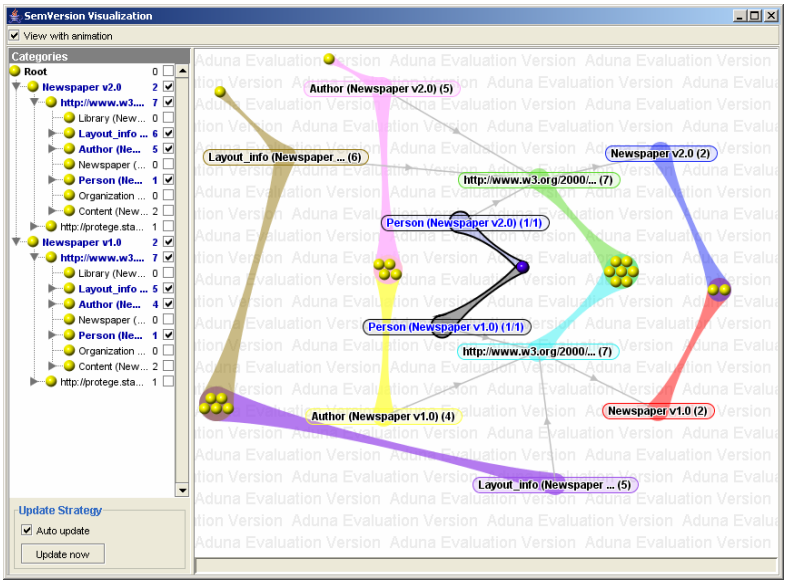

Fig. 2. Structural diff visualization in terms of classes and subclasses

SUMMARY

Versioning support for ontologies is crucial especially in dynamic environments. We presented here a methodology for RDF-based versioning that separates the management aspects from the versioning core functionality. By integrating it into Protégé, we wanted to provide the end user with the extra functionality needed to follow the normal ontology life-cycle. We intend in the future to correct the current limitations and to advance a step forward with the versioning system, first by providing collaborative versioning functionalities, like sharing versions between several users and afterwards by developing a joint negotiation mechanism for committing new versions.

ACKNOWLEDGEMENT

Part of this work has been funded by the European Commission 6th Framework Programme in the context of the Knowledge Web - Network of Excellence project, FP6-507482 and the EU IST NEPOMUK IP – The Social Semantic Desktop, FP6-027705.

REFERENCES

Klein, M., Fensel, D. 2001. Ontology versioning for the semantic web. In: *1st Int Semantic Web Working Symp. (SWWS)*, Stanford, CA, USA. pp. 75–91

Klein, M., Fensel, D. 2002. *OntoView: web-based ontology versioning*. Technical report, Vrije Universiteit Amsterdam. Submitted, draft at http://www.cs.vu.nl/~mcaklein/papers/ontoview.pdf

Sintek, M., Decker, S. 2002. Triple - a query, inference, and transformation language for the semantic web. In: *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web*, London, UK, Springer-Verlag. pp. 364–378

Staab, S., Studer, R., eds. 2004. Handbook on Ontologies in Information Systems. *Int Handbooks on Information Systems*. Springer

Stojanovic, L. 2004. *Methods and Tools for Ontology Evolution*. PhD Thesis, University of Karlsruhe.