

Ontology Change Management in Protégé

Yaozhong LIANG, Harith ALANI, Nigel SHADBOLT

Intelligence, Agents and Multimedia Group
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton, SO17 1BJ, United Kingdom
{y1504r|ha|nrs}@ecs.soton.ac.uk

Abstract. Ontology schemas tend to change and evolve over time to meet new requirements. This change may invalidate dependent applications if there is no dynamic adaptation to the changes made to underlying ontologies. Protégé, as a popular ontology development tool, should meet the challenges addressed by the evolving ontology. In this paper, we will briefly analyse the current ontology-change management in Protégé, and propose some extensions to facilitate change traceability by external application and services.

1. Ontology Management

Ontologies are becoming the corner stone of the Semantic Web (SW). Ontologies often continue to evolve over time due to domain changes, adaptations to different applications, changes to our conceptualisation or understanding of a domain, etc. Unfortunately, appropriate tools for managing ontology evolution efficiently are still missing [4]. Ontology versioning provides the mechanism to store and identify various versions of the same ontology and to highlight their differences. Ontology comparison helps ontology engineers to locate changes between different versions of an ontology [1]. If conceptual relations between various versions are built up, it becomes possible to re-interpret the data and knowledge under the different versions so that incompatibility caused by the changes made to the ontologies could be resolved semantically [2].

The lack of dynamic response to changes made to underlying ontologies may hamper the effective use of those ontologies by dependent applications. For such applications to recover and update their calls to the changed ontology, they might need to access some information that represents those changes and update accordingly.

2. Change Facilities in Protégé

Protégé evolved over the years from the role to reduce the knowledge-acquisition

bottleneck in constructing knowledge bases to an extensible, customisable, and more general-purpose toolset for constructing knowledge bases (KBs) and developing KB-based applications [3]. As ontology development became a more ubiquitous and collaborative process, to maintain Protégé's robustness in future ontology-related applications, support for ontology management becomes essential and necessary. This support should allow the ontology developers to examine and understand the changes and their rationale, and enable any dependent applications to remain up to date and compatible with the latest version of the ontology.

The following sections will describe the current change facilities supported by Protégé. In section 3 we will give out a set of requirements for a more Semantic Web friendly change management in Protégé.

2.1 Undo/Redo

Like many other development tools, Protégé has an undo/redo function which allows users to re-track their previous steps. In addition, Protégé has a *Command History* menu item, which is used to record each of the change actions the user has made to the ontology. The *history* records are timely ordered according to when the change took place, and will be deleted once the Protégé project is closed.

2.2 Ontology Version Archiving

This function provides an easy interface for users to manage different versions of an ontology. It allows users to save the current version of an ontology and add any comments to it. The saved versions will be put into separate directories with different time stamps. Protégé users can revert to any saved version using *Revert to a Previous Version*.

2.3 PROMPT

PROMP is a plug-in suite for Protégé used to manage multiple ontologies. It has four main functions:

- **Compare** the current ontology to a different version of the same ontology;
- **Move** frames between the current including project and one of the included projects;
- **Merge** two ontologies and added the resulting merged ontology to the current project;
- **Extract** a portion of another ontology and add it to the current project.

The compare function is the most relevant to the subject of this paper. Within compare, there are two kinds of views; Tree View and Table View. In the Tree View, ontology structure is displayed and the changes are highlighted. This view

aids the ontology engineers to accept/reject the changes made to the ontology. The Table View enables users to save the change record as an output file (text file) for further use.

By enabling the "journaling" preference in the system (*Project/Configure.../Options*), some information about the changes made to the ontology (e.g. author, time, item changed) will be saved to a file with the extension '.pjrn'. PROMPT is able to read such files and retrieve some information about those changes.

3. Semantically Represented Ontology Change

One of the important functionalities related to change management is logging. Logging changes is necessary to track any modification applied to the ontology. Protégé offers some limited support to ontology change logging as described earlier. Even though this logging mechanism is useful for Protégé services such as PROMPT, it can not be easily used by external services as it is not written in a machine understandable format. This is due to the reliance on Protégé's API call names in the log file, which can not be interpreted by non- Protégé programmers. Our aim is to be able to log ontology changes in a format that can be parsed and understood by general ontology-based application to allow such applications to track those changes and perhaps update their communications with the ontology accordingly.

3.1 Semantic Logging

A semantic representation of ontology change logs may enable other developers and tools to process and understand the evolution history of an ontology [6]. The type of changes that ontology engineers tend to make is highlighted in [5] and [6]. Other work investigated how best to index changes in medical terminologies to ensure their traceability [7] [8].

In our approach, we envisage a logging file to include the following aspects:

- Creating a system log ontology of change for each project;
- Populating the log ontology with instances to represent the details of any changes applied to the ontology;
- Encouraging ontology developers to write change rational and comments;
- Providing a facility to reconstruct the actions that led to the current state of the ontology;
- Providing access to the logging ontology to external tools and applications.

References

- [1] NOY, N.F., MUSEN, M.A., *Ontology Versioning in an Ontology Management Framework*. IEEE Intelligent Systems, Vol. 19, No. 4, July – August, 2004
- [2] KLEIN, M., FENSEL, D., *Ontology Versioning on the Semantic Web*, Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA, July 30 -- Aug. 1, 2001
- [3] GENNARI, J.H., MUSEN, M.A., FERGERSON, R.W., GROSSO, W.E., CRUBEZY, M., ERIKSSON, H., NOY, N.F., TU, S.W., *The evolution of Protégé: an environment for knowledge-based systems development*. In: International Journal of Human-Computer Studies, 2003, 58 (1) pp89-123
- [4] SURE, Y., *On-To-Knowledge -- Ontology based Knowledge Management Tools and their Application*, In: German Journal Kuenstliche Intelligenz, Special Issue on Knowledge Management (1/02), 2002
- [5] KLEIN, M., KIRYAKOV, A., OGNJANOV, D., FENSEL, D., *Finding and Characterizing Changes in Ontologies*, The 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7 - 11, 2002
- [6] MAEDCHE, A., MOTIK, B., STOJANOVIC, L., STUDER, R., VOLZ, R., *Managing Multiple Ontologies and Ontology Evolution in Ontologging*, Intelligent Information Processing 2002, pp51-63
- [7] OLIVER, D. E., SHAHAR, Y., *Change Management of Shared and Local Health-Care Terminologies*, IMIA Working Group 6 Meeting on Health Concept Representation and Natural Language Processing, AZ, U.S.A, 1999
- [8] CIMINO J.J., *Formal descriptions and adaptive mechanisms for changes in controlled medical vocabularies*, Meth Inform Med, 1996 Sep, 35(3) pp202-10