Abstract: Graph Visualisation to Aid Ontology Evolution in Protégé

Benjamin Blundell and Steve Pettifer
Advanced Interfaces Group (AIG)
Manchester University, Computer Science Department
Oxford Road, Manchester
blundell@cs.man.ac.uk

1. The Current Problem

Ontology evolution is one of the key problems facing ontology users today. Adapting ontologies to meet new requirements involves understanding various sections of ontologies and the changes made thereafter. Currently, a large amount of research has been undertaken in automatic change detection and change effects, however the process is far from automated. Users are involved at almost every step and with ontologies becoming more commonplace and complex, tools are required to lighten the large cognitive load. Initially, we present the requirements and needs for a visualisation solution in order to streamline ontology evolution, followed by the research into a graph based prototype, integrated with Protégé.

2. Summary of Visualisation Requirements

The following list of requirements are applicable to visualising changes to ontologies. They have been derived from papers and interviews with users. In addition, we have approached the problem from a second angle; that presented by Gary Ng [6] which is discussed later.

1. Representation of the data transformed between two ontology versions[3].
2. Informing the user of Axiom validity across changes[3].
3. Distinguish Semantic and Syntactical changes[2][9]
4. Highlight potential specification violations resulting from semantic changes[2].
5. Support Basic and Complex change distinctions / Abstraction[3].
6. Ability to vary the level of granularity of changes / Group changes[8].
7. Present possible implications and cascades[8].
8. Present data in an orderly fashion, reducing unnecessary detail, particularly in large ontologies. Related to scope and abstraction[9, 8, 3].
9. Ability to reverse changes and review previous ontology versions[9].
10. Identify terms which may become redundant due to changes.
11. Differentiate between inferred and actual links. A Reasoner issue.
12. Viewing the ontology with respect to different link types.
13. Identifying the term migrations performed by the reasoner.
14. Explore the view interactively in order to discover further ramifications.
15. Differentiating between original and imported classes / ontologies.

As mentioned at the beginning of the section, the approach given in [6] was adopted in addition to interviews and collating papers. The work undertaken is rather extensive; a summary of which is presented here.

As a result of this approach, we have identified three major high-level task comprising of smaller visual tasks with various scopes and user approaches. These tasks are presented in table 1.

The first high level task consists of two subtasks. Distinguish refers to maintaining the notion of place whilst presenting the important data, whilst Locate refers to actually finding the data in question.

Comprehension is broken down into 3 tasks. Reveal is simply the act of understanding what the given picture means, and is born from the other two tasks. Generalise is a task that requires special consideration. Often, users have expressed a wish to view ontologies and changes to said ontologies at different levels of granularity. Related to the Distinguish task above, generalize involves presenting the knowledge identified in the previous task in such a

High Level Task	Scope	Visual Tasks
Navigation and Initial identification	Intermediate, global	Locate, Distinguish
Further identification and comprehension	Elementary / Intermediate	Generalise, Emphasize, Reveal
Comparison between states	Elementary / Intermediate	Associate, Compare, Reveal

Table 1.

way as to be meaningful on various levels. It is desirable to show changes both as atomic, basic changes and more semantically meaningful complex changes. Emphasize is another visual task which brings forward the notion of context. Users have suggested being able to view ontologies with respect to certain relationship types or classes. Bringing a particular relationship type to the fore, emphasizes a certain aspect of the information which the user might be interested in.

Finally, comparison between states has three subtasks. Again, a reveal task to cover the actual comprehension of the successive images, and 2 further tasks supporting the reveal. Associate deals with the fact that the items involved in the changes must be identified and linked. For example, a class may appear different after one change, and thus the system should associate these two stages for the user. This enables the Compare task. The system should again, reduce the users cognitive load by presenting the change in a way that the user can easily understand.

3. Current Work

As ontology evolution is a large and complex area of ontological development, we have decided to initially focus on the differences presented between inferred and asserted hierarchies; a common problem in the development of ontologies. Although there exists a reasonable number of visualisations for ontological development, the majority do not satisfy the list of requirements we have presented; indeed, very few are tailored to any specific problem within the ontology domain.

We have developed a graph based plugin, similar to OWLViz only displaying not only the subsumption hierarchy, but all relations between the different classes thus reflecting requirement 12. Indeed, our plugin veers away from the classic ”subsumption hierarchy” as the majority of ontologies are not hierarchical when relationships other than subsumption are considered.

4. Discussion

As graphs are the classical way of presenting relational data, the problem becomes how to extend their expressive nature to indicate changes within a succession of graphs over time. A simple approach is to use ”tweening style” animation as a cue, paying attention to the Gestalt principles laid out in [5]. A considerable amount of time has been dedicated to selecting an appropriate representation and layout algorithm as each particular approach has many benefits and drawbacks. Currently, we are using a modified force placement algorithm, coupled with a Pre-processor [4] to increase performance. Finally, to minimize potential random placements between graphs, the algorithm has been extended to three dimensions, taking advantage of a technique similar to multi-dimensional scaling.

An interesting approach by [1] helps to preserve the ”mental map” of the graph by performing a transformation between two states using affine, linear transformations. Outliers can then be highlighted and brought to the users attention. By utilising the third dimension, we can adapt the approach given in [7] to support interactive feature discovery.

A large body of work exists regarding automatic detection and discovery. [3] presents PromptDiff and several other textual based programs along with a potential standard for ontology change recognition with OWL. Merging such a methodology with a potential visualisation package presents interesting questions.

We would like to present and discuss the requirements laid out for visualisation, the need for aiding ontological evolution through graphical means, and our initial prototypes. It is our belief that visualisation of changes may greatly aid ontological engineering in the future.

5. Acknowledgements

The authors would like thank Carole Goble, Matthew Horridge, Mike Bada and Nick Drummond, of the Information Management Group (Manchester University), for their assistance in identifying some of the current problems facing ontology users.

References

[1] C. Friedrich and P. Eades. Graph drawing in motion. *Journal of Graph Algorithms and Applications*, 6, 2002.

[2] M. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*, 2002.

[3] M. Klein and N. F. Noy. A component-based framework for ontology evolution. In *Proceedings of the Workshop on Ontologies and Distributed Systems, IJCAI'03*, 2003.

[4] P. Mutton and P. Rodgers. Spring embedder preprocessing for www visualization. In *Proceedings of Information Visualisation 2002 (IV02)*, 2002.

[5] K. V. Nesbitt and C. Friedrich. Applying gestalt principles to animated visualizations of network data. In *Sixth International Conference on Information Visualisation (IV'02)*, page 737, July 2002.

[6] G. Ng. *Interactive Visualisation Techniques for Ontology Development*. PhD thesis, University of Manchester, Computer Science Department., November 2000.

[7] G. Robertson, K. Cameron, M. Czerwinski, and D. Robbins. Polyarchy visualization: Visualizing multiple intersecting hierarchies. *Information Visualization*, 1, March 2002.

[8] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven ontology evolution management. In *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*, 2002.

[9] L. Stojanovic and B. Motik. Ontology evolution within ontology editors. In *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*, volume 62, September 2002.