Visualising a DL Knowledge Base with DeLogViz

Bo Hu and Nigel Shadbolt
Department of Electronics and Computer Science
University of Southampton
United Kingdom
Email: {bh, nrs}@ecs.soton.ac.uk

Abstract

Ontologies are more than an elaborate approach to the definition and classification of information. They play a vital role in various web-oriented applications that makes a correct ontology one of the first priorities. In this paper, we present an ontology visualising and authoring tool which is enhanced by an interactive graphic user interface. With direct visual interpretation, ontology authoring can be more convenient and accurate.

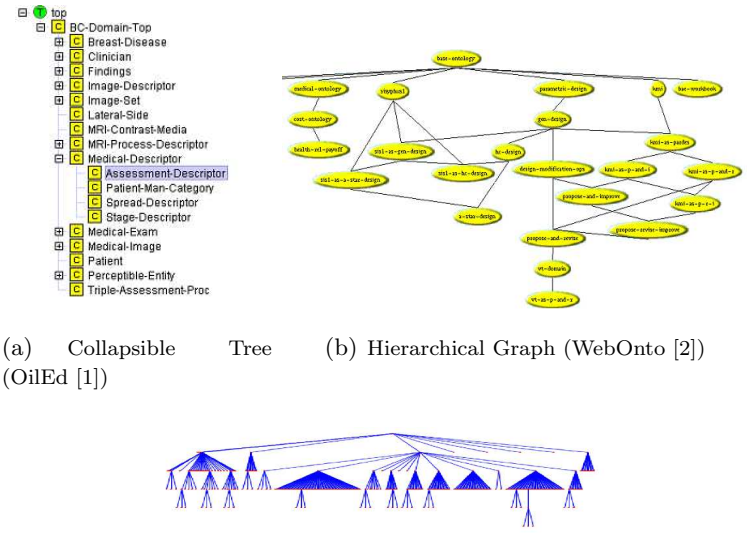
1 Introduction

Currently, most ontology modelling languages aiming at facilitating web applications are fully compliant with the XML standards [9]. In order to tackle the ever-growing size and complexity of the web and enable the sharing and reuse of knowledge without dealing with XML tags, various editing tools have been proposed and implemented with the promises to simplify the construction of ontologies, e.g. OilEd [1], Protégé [7] *etc.* Such promises are even more important for tools targeting end users who are domain experts (e.g. medical staff) with limited training in knowledge representation and computer programming.

In this paper, we describe a new ontology visualising and authoring tool, DeLogViz, that integrates an interactive visualisation user interface with a Description Logic (DL) based knowledge base management system. The motivation behind DeLogViz is to enable domain experts, with limited knowledge of computer science and artificial intelligence to maintain and populate the knowledge base relatively easily.

2 Ontology Visualisation

Visualising and navigating information is a challenging problem faced in many knowledge domains. It is admitted that the same information can be viewed from different perspectives with different outcomes. Research in cognitive science reveals that the format in which information is presented to users has a fundamental impact on the ways in which users perceive and interpret the information and thus affects their judgement and performance based on the information [6]. Evidence also shows that, in many cases, graphical representations are received and interpreted more efficiently than textual ones [6].



(a) Collapsible Tree (OilEd [1])

(b) Hierarchical Graph (WebOnto [2])

(c) Hierarchical Graph View of Very Large KB

Figure 1: Traditional Style for DL-enable Editor

Available ontology authoring tools use either a collapsible tree view (Figure 1(a)) or a hierarchical graph view (Figure 1(b)), with native code or via plug-ins, to browse the knowledge structure. Among them, only a few tools provide graph-based editing[1]. Strictly speaking, collapsible trees are not considered as graphs and thus restrict the information can be delivered. Although a tree view gives direct visual information on the hierarchical structure, it is not expressive enough to present all the relational information. For instance, in most cases, only the subsumption relationships are expressed in a collapsible, direct manipulation tree, while information regarding individuals or role referencing relations is left to textual representation. On the other hand, a hierarchical graph provides more expressive power at the price of scalability and space efficiency [4]. As shown in Figure 1(c), when visualising a very large ontology, it could be very difficult to distinguish different nodes and edges.

The lack of ontology visualising tools able to handle large graphs gave us a reasonable inspiration to implement DeLogViz which takes full advantages of a hyperbolic graph view [5] to efficiently deploy large knowledge bases.

3 Visualised DL Knowledge Base

The characteristics of knowledge presented in a DL knowledge base matches very well with those of a 2D directed graph. While nodes of the graph map to concepts and individuals in the knowledge base, relationships among concepts (and/or individuals) are mapped to edges.

In a DL knowledge base, relationships can be grouped in two categories, i.e. the *is-a* relationship: subsumption and instantiation, and the role referencing relationship. Such differences are literally represented in DeLogViz by edges with different colours and different labels (Figure 2). Meanwhile, arrows are used to indicate the direction of edges and thus reflect the direction of the *is-a* or role referencing relationships.

[1] A thorough survey of ontology editing tools is available online from http://www.xml.com/2002/11/06/Ontology_Editor_Survey.html.

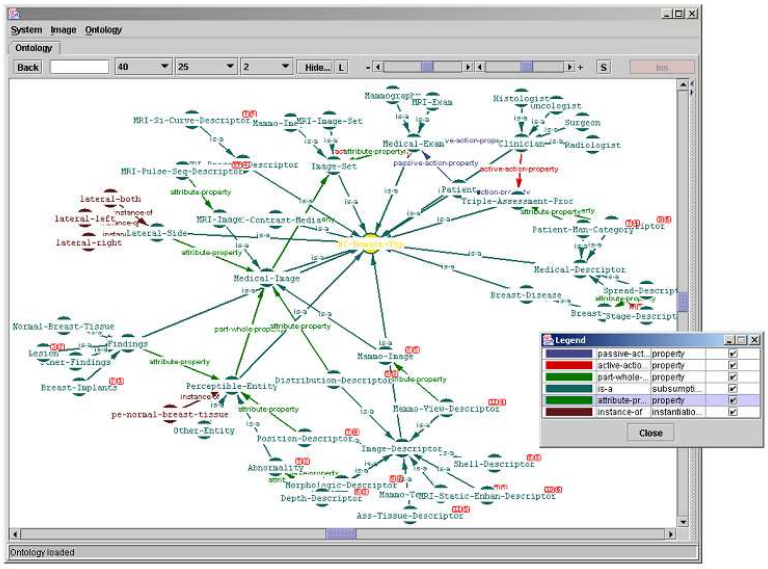


Figure 2: Browser DL Knowledge Base

3.1 Visualising

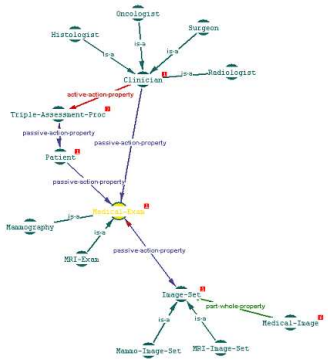
In DeLogViz, the deploy of nodes and edges is managed by TouchGraph [8], a Java application designed to visualise interrelated networks using hyperbolic-like interactive graphs. It provides various navigation and interaction facilities, e.g. zoom and pan, and "fisheye distortion" (focus+context zoom) (Figure 3), which are essential for visualising large and complex graphs [4]. Such functionalities are enhanced in DeLogViz by allowing an optional "properties-of-interest" to be visible or hidden. This is especially beneficial when visualising a knowledge base where different types of edges may exist between a pair of nodes.

DeLogViz marks each node with two complexity cues: the number of edges going into or away from the current node which is not shown on the graph; and the number of nodes in a subgraph of which the current node is the root, if only the edges representing subsumption relationships are considered. Such functionality allows a subgraph to be collapsed to a node which helps reduce the graph complexity, yet does not prevent such complex information being disclosed to users.

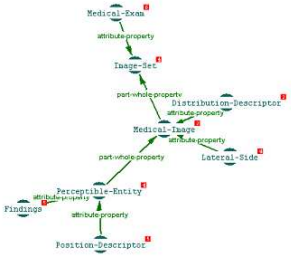
3.2 Editing and Reasoning

Knowledge base can be modified by editing directly on the graph (e.g. creating or deleting nodes) which brings up an input form. DeLogViz uses a Database-like input form for knowledge acquisition which is under the assumption that, to some extent, the usability of such an approach is high because of the familiarity of domain experts with the input style of the traditional relational database systems. However, such an assumption needs further investigation with user-based experiments.

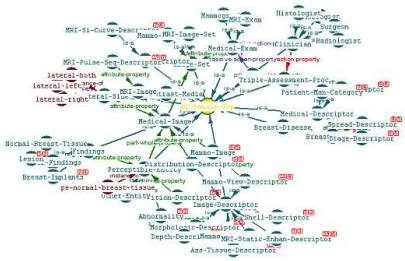
Input knowledge is stored in a format compliant with DAML+OIL syntax. A DL reasoning engine is invoked, if appropriate, to provide consistency checks and updates of the original taxonomy. The latest updates are then stored as a file on the disk to allow off-line retrieval of the nodes and relations contained in the ontology.



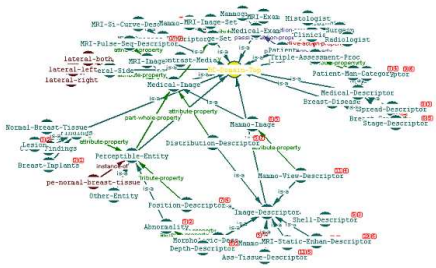
(a) Relocating the focus to a particular concept



(b) Displaying only the properties of interest



(c) Before Fisheye-Distortion



(d) After Fisheye-Distortion

Figure 3: Effects of navigation and interaction facilities

4 Discussions and Future Work

We have presented DeLogViz, which integrates traditional ontology authoring tools with an interactive visualisation user interface. It enhances the understanding of ontologies through visual perception. However, as pointed out in previous sections, we have not definitively established whether or not using a hyperbolic graph view improves the usability of a DL-based modelling language, although it is assumed that information will be better received with visual representation. A controlled evaluation of systems with interactive graph visualisation and traditional methods of visualisation needs to be carried out.

DeLogViz also benefits from a database-like knowledge acquisition interface. We expect such an approach will be of help when maintaining and populating an ontology is in the hand of the domain experts. DeLogViz is considered as a tool for visualising information in the hope that, together with textual representation, the visualised structure can help to achieve a better comprehensibility of ontologies.

DeLogViz is not yet fully-fledged. It is expected to be improved on the following two aspects.

DL reasoning server : In the current implementation, DeLogViz connects to the RACER [3] server for DL-based reasoning. It is expected that other reasoning servers may be considered to improve the system flexibility and interoperability. Meanwhile, essential queries to the DL reasoning server (e.g. retrieving all subclasses of a named class) are expected to be fully covered by buttons, menu items or other graphical resources. The consequences of such improvement is not

trivial as we expect some end users to be domain experts with limited knowledge on logic programming rather than highly skilled knowledge engineers. Fulfilling complex functionalities by a single click of the mouse will encourage the use of our system.

Database server : DeLogViz uses temporary working files to communicate between different components and for re-graphing the updated taxonomy after DL reasoning. This solution may be replaced with an RDF triple store facilitated by MySQL or other database servers. The translation from RDF back into DAML+OIL is enabled by XSLT rules. "Serveriseing" the storage of the knowledge base can also facilitate internet access to DeLogViz.

Acknowledgements

This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under the MIAKT grant GR/R85150/01. The authors are grateful to Dr. Harith Alani for providing valuable inspirations and insights about the Knowledge Base visualisation.

References

[1] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: a reason-able ontology editor for the semantic web. In *Proceedings of KI2001*, number 2174 in Lecture Notes in Computer Science, pages 396–408. Springer-Verlag, 2001.

[2] J. Domingue. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In *Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1998. Banff, Canada.

[3] V. Haarslev and R. Möller. *RACER User's Guide and Reference Manual Version 1.6*. University of Hamburg, Computer Science Department, July 2001.

[4] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[5] J. Lamping, R. Rao, and P. Pirolli. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*, pages 401–408. ACM, 1995.

[6] J.H. Larkin and A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–99, 1987.

[7] Stanford Medical Informatics, Stanford University. *Protégé-2000*. Copyright 2003, Available from http://protege.stanford.edu/.

[8] TouchGraph LLC. *TouchGraph*. Available from http://www.touchgraph.com/.

[9] World Wide Web Consortium. *Extensible Markup Language (XML)*. Available from http://www.w3.org/XML/.