

Selective Ontology Viewer

Krishnakumar Pooloth, Swaminathan Natarajan
Software Engineering and Technology Labs,
Infosys Technologies Ltd,
Electronics City, Hosur Road,
Bangalore 560100,
India

Phone : +91-80-28520261

Fax : +91-80-28520740

EmailID: {krishnakumarp,swaminathan_n01}@infosys.com

Abstract

As the value of ontology based information systems is beginning to be realized, the mechanism of visualizing ontologies assumes importance. For small ontologies, it might be easy for the user to browse the full ontology and find relevant information. Special visualization techniques are required for large ontologies to enable the user to find the relevant information quickly. In this paper, we present a new technique for visualizing relevant sub-sections of large ontologies. In our approach, a user defines a sub-section of ontology structure as a template. Specific instances can be visualized based on this template.

Introduction

As the enterprises are realizing the value of ontology repositories and ontology based applications, issues in visualization of ontologies assumes importance as a key area of research. Effective procedures or mechanisms for visualization of ontologies would greatly facilitate the adoption of ontology based solutions in the business computing arena.

Protégé [1] is a very popular Ontology Editor developed at Stanford University. Ontologies can be edited as well as viewed using Protégé. It also provides a Java API to access the ontologies. Its plug-in architecture allows for extension of the tool with new features and capabilities.

In this paper, we present "Selective Ontology Viewer" (SOV plug-in), an ontology visualization plug-in for Protégé. The plug-in uses JGraph [2], an open source graph visualization library. The plug-in is light-weight and lets the user select parts of ontology structure as templates and visualize instances based on this template.

Existing Visualizers

In this section we will briefly look at existing ontology visualization plug-ins for Protégé.

OntoViz[3], based on AT&T's GraphViz software, provides customizable graphical visualizations of Protégé's ontologies. OntoViz generates graphs with very good clarity with little overlap between nodes. However, OntoViz graphs are static and non-interactive, making them non-suitable for certain purposes.

Jambalaya[4] is based on ShriMP[5]. ShriMP is an information visualization technique for navigating abstracted structures of complex information networks with animated zoom. Jambalaya plug-in integrates ShriMP into Protégé. It offers interchangeable views of nested graphs for interactive ontology navigation. Jambalaya supports wide range of layout algorithms such as Tree layout, Radial layout etc.

TGVizTab[6] is another visualization plug-in for Protégé which uses TouchGraph[7] for graph infrastructure. It supports handling different types of relations and edge

labeling. It allows for storing the generated graph in XML which can be viewed by other TouchGraph applications.

Selective Ontology Viewer

Often, users of a large ontology would want to look at it from different perspectives. For e.g., in newspaper ontology a user might want to look at reporting hierarchy of an Editor, or the contents in an instance of Newspaper etc. Selective Ontology Viewer was developed to help users easily view such specific relationships quickly. In SOV plug-in, the user defines a set of frequently required templates from the ontology structure. The template is a sub section of the ontology structure starting with a root class which is of primary interest for the template. This template is stored permanently as an XML file. Later, the user can visualize specific instances based on this template. The graph visualization is achieved using an open source Java graph library, JGraph¹.

SOV plug-in Features

Figure 1 shows the user interface of Selective Ontology Viewer. The currently defined ontology templates are displayed on the left and visualized templates are shown on

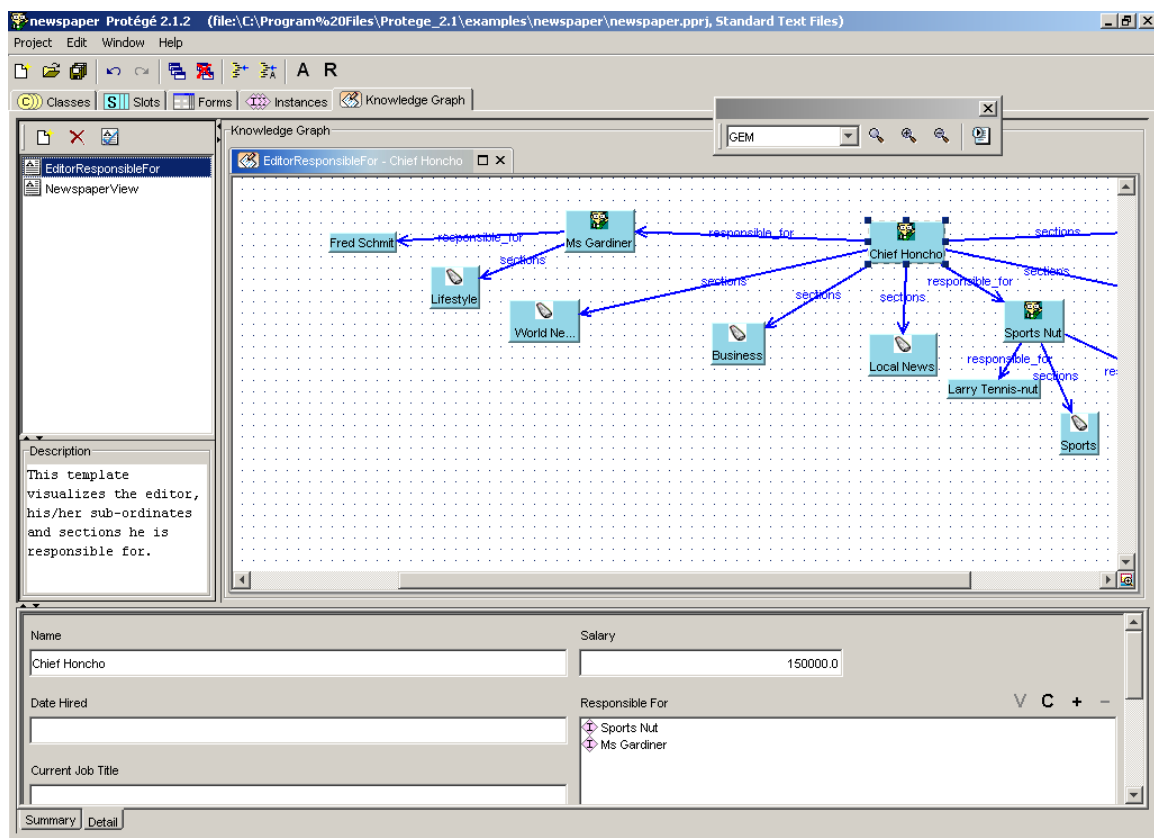


Fig 01: SOV User Interface

¹ JGraph is the core graph visualization library of the JGraph suite. The main advantage of JGraph is that it is completely compliant to Java Swing API. JGraph is written in pure Java and provides key features such as zooming, cell collapsing/expanding, undo, event-handling, drag and drop support etc

the right. Below, we elaborate on the features supported by the visualizer.

Selective Viewing based on templates

The main aspect of SOV plug-in is the capability to define templates which can be reused. The SOV templates can be defined as a sub-graph of ontology structure with a class being the sole root node. The graph could have infinite depth by traversing the slots of the root class which are of Instance Type. In other words, user selects specific paths in the ontology structure starting with a class as the root node. Figure 2 shows the user interface for specifying a template.

Persistence of visualization templates

The SOV plug-in specifies an XML format for the templates and allows for saving of these templates so that they can be used across Protégé user-sessions. This enables easier usage compared to other plug-ins, since you don't have to set up the visualizer each time you use it.

This template can be used to visualize specific instances of the root class of the template. In the generated graph, instances would be displayed as nodes and the slots that specify the relationships between these instances would be represented using directed edges. For e.g., the relationship between an Editor and Reporter instance would have the instances as nodes and the slot "responsible_for" as a directed edge between the two. The Protégé's Instance view of any instance in the graph can be seen by selecting the corresponding node. This view would be shown in the bottom panel of the plug-in.

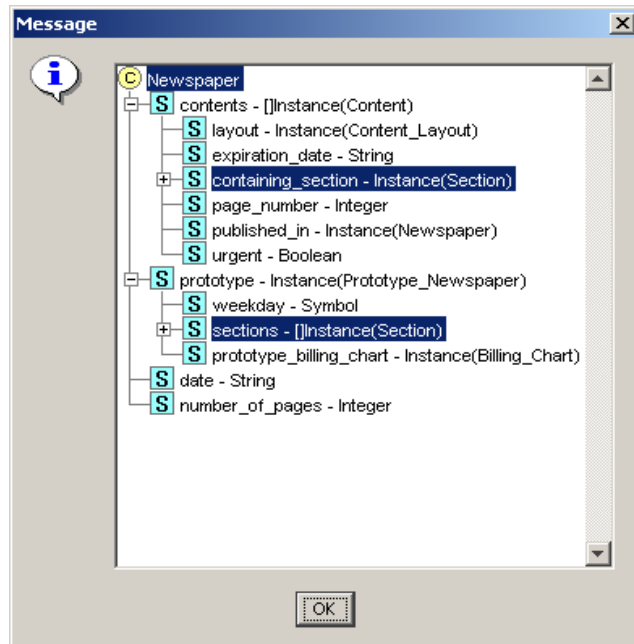
The plug-in supports visualization of recursive relationships. For e.g., in Newspaper ontology if there is reporting relationship between Editor A and B, the plug-in would automatically expand the reporting hierarchy of Editor B as well.

Mapping icons to Classes

The plug-in also allows the user to customize graph by mapping classes to icons. The icon would appear for the nodes if the class corresponding to the instance has been mapped to an icon. This can be used to associate specific icons classes such as Reporter, Editor, Section etc. Appropriate usage of image would render the graph more readable and easy to use.

Layout Support

The plug-in supports multiple options for graph layout based on common algorithms such as Sugiyama, GEM, Simulated Annealing etc. This is implemented by leveraging the support for layout algorithms in JGraph.



Easy Scrolling using graph miniature

The user can click on the corner button of the scroll pane that contains the graph to see a miniature version of the graph. The miniature can be used to easily scroll the graph and find required information faster.

Export as HTML

Using SOV plug-in, a user can save the graph and the corresponding instances as an HTML file for offline viewing.

Conclusion

Easy and customizable visualization mechanisms would help in making the knowledge encoded in an ontology more accessible to end-users. In this paper we described Selective Ontology Viewer; a new Protégé visualization plug-in built using JGraph. It allows for explicit definition of templates which are sub-graphs of the ontology structure and visualization of instances based on these templates.

References

- [1] Protege-2000, The Protégé website, Stanford Medical Informatics, <http://protege.stanford.edu>
- [2] JGraph, The JGraph website, <http://www.jgraph.com>
- [3] Sintek, M., OntoViz Tab: Visualizing Protégé Ontologies, <http://protege.stanford.edu/plugins/ontoviz/ontoviz.html>
- [4] Storey, M.-A.D., Musen, M.A., Silva, J., Best, C., Ernst, N., Fergerson, R. and Noy, N.F., Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protege. In Workshop on Interactive Tools for Knowledge Capture, K-CAP-2001, (Victoria, B.C. Canada, 2001).
- [5] Storey, M.A., Wong, K., Fracchia, F.D., Muller, H.A.: On Integrating Visualization Techniques for Effective Software Exploration. Proc. IEEE Symp. on Information Visualization (InfoVis'97), Phoenix, Arizona, USA. 38-45, 1997.
- [6] Alani, H., TGVizTab: An Ontology Visualisation Extension for Protege. in Knowledge Capture 03 - Workshop on Visualizing Information in Knowledge Engineering, (Sanibel Island, FL, 2003), ACM, 2-7.