Web-Protégé – Protégé going Web

Tania Tudorache, Jennifer Vendetti, Natalya F. Noy

Stanford Center for Biomedical Informatics Research, Stanford University, CA, US
{tudorache,vendetti,noy}@stanford.edu

Abstract. In this presentation, we will talk about Web-Protégé—a web-based lightweight ontology editor. Web-Protégé is open source, uses the Google Web Toolkit (GWT) for the user interface, and Protégé for supporting ontology services. We used components of Collaborative Protégé to augment the ontology-editing environment with facilities for discussions and annotations. In this paper, we describe both the server and the client components of the system. We built the user interface using the portal metaphor, which allows easy customization of the user interface for individual users. The plug-in architecture allows easy extension of the user interface with custom components. The server component is implemented in Java and provides methods for accessing the ontology content and manages the changes the users make in different clients. A demo version is available at http://bmir-protege-dev1.stanford.edu/webprotege/.

1 Introduction

With the advent of Web 2.0 technologies and applications, the web has become the primary environment where people communicate, exchange content, and collaborate on projects. Meanwhile, ontologies are becoming so large in their coverage that no single person or a small group of people can develop them effectively and ontology development becomes a community-based enterprise. In this situation, the Web becomes the natural platform of choice for developing ontologies collaboratively.

2 System description

We are currently developing Web-Protégé—an open source lightweight ontology editor for the Web that uses Protégé as its backend. Our main goal in developing Web-Protégé is to better support the collaborative development process in a web environment. In this work, we leverage the existing components that we have developed for Collaborative Protégé [2], and which already provides support for simultaneous editing (a change made by an user is immediately seen by the other users). In this context, Web-Protégé is a Web front-end for the Collaborative Protégé server. This means, that different users may edit the same ontology either in Web-Protégé or in a Collaborative Protégé desktop client and they will see each others changes right away.

Figure 1 shows a simplified version of the Web-Protégé **architecture**. The **server side** provides ontology-access services through the Ontology API (current version uses the Protégé-OWL 3.x API). This Java API contains methods for reading and writing OWL ontologies. In addition, the server component provides support for collaboration services, such as annotation of ontology components and change tracking. Users may attach *annotations*, which are typed comments (e.g., example, proposal, question, etc.),

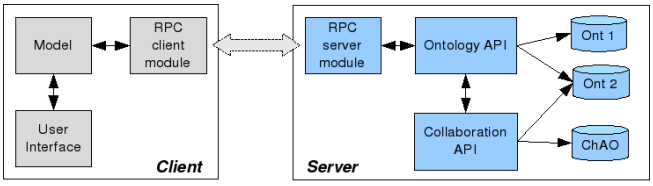

Fig. 1. Simplified architecture of Web-Protégé. The user interface interacts with a model of the ontology on the client side via a listener pattern. When the model needs to be filled with new information, the Remote Procedure Call (RPC) module on the client will invoke a request to the RPC module of the server, which interacts with the *Ontology* and the *Collaboration APIs* to provide the requested data.

to ontology components, to the ontology as a whole, to the descriptions of ontology changes, or to other annotations.

The **client side** provides the user interface, a model of the ontology on the client and the Remote Procedure Calls (RPC) module to communicate with the server. We used the Google Web Toolkit[1] (GWT) to implement the user interface. GWT allows a developer to write a web front end in Java and then compile the source code into highly optimized JavaScript.

The client has an internal model of the ontology that represents the ontology view of the client. The content of the client model is filled by user interface requests (e.g., get all subclasses of a class), and it also serves as a client-side cache. The user interface components use a listener pattern to register for changes in the client model so that they can refresh when the model changes. In the current implementation, the communication between the client and the server is made via GWT RPC calls.

Figure 2 shows the Web-Protégé **user interface**. In designing the user interface, we took inspiration from well known portals, such as MyYahoo and iGoogle. We refer to each component in the user interface as a *portlet* (e.g., Class tree, Notes etc.). Users can easily customize the appearance of the interface using drag-n-drop. Users can also show or hide specific tabs, or add other portlets to a tab via toolbar buttons. We have implemented a plug-in architecture for the tabs and portlets, so that developers can easily plug their own components into the user interface of Web-Protégé.

3 Discussion and Summary

We have presented Web-Protégé—a lightweight ontology editor for the Web with support for collaboration. Two groups developing biomedical ontologies have used Web-Protégé and have found it useful The current version uses Protégé 3 as its backend and only supports OWL 1.0. In the near future, we will also support OWL 2.0 by implementing a backend using Protégé 4 and the OWL-API. [2] We plan to use a more flexible architecture for managing ontology changes that will provide a pluggable way of solving conflicts during edit time. [1] We will extend the support for a wider range of collaboration features by exposing change tracking information in the UI, enforcing access policies for browsing and editing ontologies, and supporting different types

[1] `http://code.google.com/webtoolkit/`
[2] `http://owlapi.sourceforge.net/`

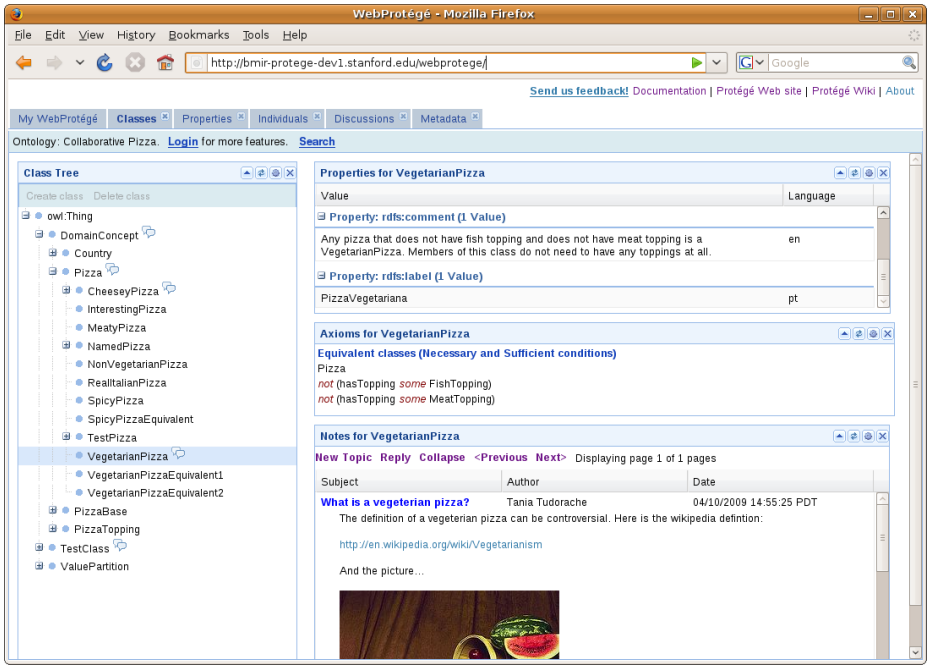

Fig. 2. User interface of Web-Protégé. The figure shows a screenshot of the NCI Thesaurus in Web-Protégé. The user interface is made up of UI components, called portlets (e.g., Class Tree, Notes, etc.). The user can customize the layout and appearance of the UI by drag-n-drop, and by showing and/or hiding portlets.

of collaborative workflows. Scalability is also an issue for large ontologies due to some limitations in the use of JavaScript. We plan to develop widgets with pagination support that would better handle large ontologies.

Acknowledgments

We are indebted to Dilvan Moreira for suggesting the use of GWT. Protégé is a national resource supported by grant LM007885 from the U.S. National Library of Medicine.

References

1. T. Redmond, M. Smith, N. Drummond, and T. Tudorache. Managing change: An ontology version control system. In *OWL: Experiences and Directions, 5th Intl. Workshop, OWLED 2008*, Karlsruhe, Germany, 2008.
2. T. Tudorache, N. F. Noy, and M. A. Musen. Supporting collaborative ontology development in Protégé. In *Seventh International Semantic Web Conference, ISWC 2008*, Karlsruhe, Germany, 2008.