Web-Protégé: A Lightweight OWL Ontology Editor for the 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 paper, we present 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. For example, the National Cancer Institute (NCI) that develops the NCI Thesaurus—an OWL ontology with more than 80.000 concepts—"outsource" some of their ontology development to the scientific community at large using a wiki as a collaborative platform. Other biomedical projects use a combination of available tools (e.g., sourceforge, mailing lists, etc.) due to the lack of proper support for web-based ontology development.

We interviewed several groups of domain experts and ontology engineers who are building ontologies collaboratively and extracted some of the main requirements for a web-based collaborative ontology environment. First, users need a simple way of making their ontology available on the web, so that other people can browse it without the need to install software. Second, users want to comment on or discuss certain aspects of the ontology—with such discussion as an integral part of the development process. Third, users want to make changes to the ontology in a web browser, and still get the same sorts of editing help available in desktop-client editors (e.g., validation of user input). Several groups expressed the need for a simplified view of the ontology that is published on the web, so that they don't "scare off" users who are not ontology experts. Therefore, being able to customize the appearance of the web interface based on the level of expertise of the user becomes a crucial feature. In the following sections, we describe how Web-Protégé addresses some of these requirements.

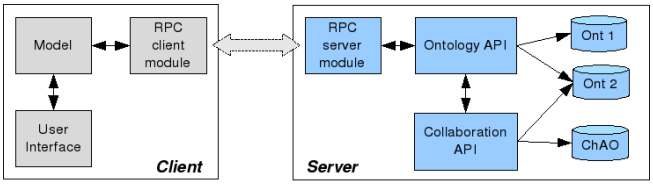

Fig. 1. Simplified architecture of Web-Protégé. The user interface interacts with a model of the ontology on the client side via a listener pattern. When the model needs to be filled with new information, the Remote Procedure Call (RPC) module on the client will invoke a request to the RPC module of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

2 System description

Our laboratory has developed Protégé[1]—a widely used open-source ontology and knowledge base editor. Collaborative Protégé [3] is a Protégé extension that enables users who develop an ontology collaboratively to hold discussions, chat, annotate ontology components and changes—all as an integral part of the ontology-development process.

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 already developed for Collaborative Protégé [3], and which already provide 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.

Our plan is to provide a collaboration platform that is easily customizable for different users and projects' settings. For example, a novice user may use Web-Protégé to browse and do lightweight editing of an ontology (e.g., move or rename classes); a more advanced user may connect to the same ontology with a Collaborative Protégé desktop client to perform more complex changes (e.g., creating restrictions, refactorings, etc.). Because the two users are working with a shared ontology, they will see each others' changes immediately.

2.1 Architecture

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 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.), to ontology components, to the ontology as a whole, to the descriptions of ontology changes, or to other annotations. We define the structure of the annotations in the *Changes and Annotations ontology* (CHAO) [1]. CHAO contains classes that define the annotation

[1] `http://protege.stanford.edu`

types (e.g., *Comment*, *Proposal*) and classes for describing different types of changes that a user can do in an editing session (e.g., *Class_Created*). Instances of the annotation types will represent actual annotations that users are making on ontology components or changes. We associate each domain ontology stored in the server repository with a CHAO knowledge base that contains the instances of annotations and changes corresponding to the domain ontology.

An important server task is to keep track of the changes in the ontology and to manage conflicts when different clients make changes to the same ontology. The Web-Protégé server makes calls to the Collaborative Protégé server that already provides support for simultaneous editing, transactions, and operation atomicity. The task of the Web-Protégé server is to manage the Web-Protégé clients. For each ontology, the server maintains a current version number. Whenever a change is made to the ontology, the ontology version number is incremented by one. Each client has its own version number, which might differ from the server's version number. At a set time interval (default 5 seconds), the clients poll the server for new changes. The client will send the server its own ontology version number, and, in turn, the server will send the client the diff of changes between the client and server versions. When the client receives the diff of changes, it will update its internal model and its ontology version number.

The **client side** contains 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[2] (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.

2.2 User Interface

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. However, we are in the early stages of development and much remains to be done. The current version uses Protégé 3 as its backend and

[2] `http://code.google.com/webtoolkit/`

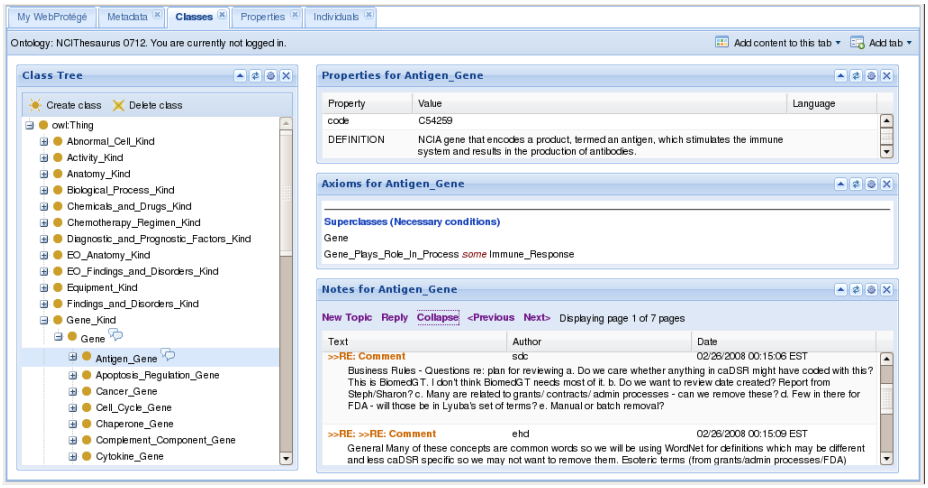

Fig. 2. User interface of Web-Protégé. The figure shows a screenshot of the NCI Thesaurus in Web-Protégé. The user interface is made up of UI components, called portlets (e.g., Class Tree, Notes, etc.). The user can customize the layout and appearance of the UI by drag-n-drop, and by showing and/or hiding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

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. [3] We plan to use a more flexible architecture for managing ontology changes that will provide a pluggable way of solving conflicts during edit time. [2] 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 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. N. F. Noy, A. Chugh, W. Liu, and M. A. Musen. A framework for ontology evolution in collaborative environments. In *5th Intl. Semantic Web Conference, ISWC*, volume LNCS 4273, Athens, GA, 2006. Springer.
2. 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.
3. T. Tudorache, N. F. Noy, and M. A. Musen. Supporting collaborative ontology development in Protégé. In *7th Intl. Semantic Web Conference, ISWC 2008*, Karlsruhe, Germany, 2008.

[3] `http://owlapi.sourceforge.net/`