

# Collaborative Protégé

Tania Tudorache  
Stanford Medical Informatics  
251 Campus Drive  
Stanford, CA 94305, USA  
tudorache@stanford.edu

Natasha Noy  
Stanford Medical Informatics  
251 Campus Drive  
Stanford, CA 94305, USA  
noy@stanford.edu

## ABSTRACT

We present an extension of the existing Protégé<sup>1</sup> system that supports collaborative ontology editing. The extended system—Collaborative Protégé—is currently available as a prototype. In addition to the common ontology editing operations, it enables annotation of both ontology components and ontology changes. It supports the searching and filtering of user annotations based on different criteria. We have implemented two types of voting mechanisms that can be used for voting of change proposals.

## Categories and Subject Descriptors

H.5.3 [Group and Organization Interfaces]: Collaborative computing, Computer-supported cooperative work, Web-based interaction; H.3.5 [On-line Information Services]: Web-based services, Data sharing

## General Terms

Design, Management, Measurement, Algorithms

## Keywords

Ontology development, Collaboration, Annotation, Change management, Protégé

## 1. INTRODUCTION

Existing ontology development environments have focused so far on the editing of ontologies in a stand-alone, single-user mode. The emerging of the Semantic Web and of the Web 2.0 technologies bring a new challenge to the ontology editor providers, the collaborative editing of ontologies in a web-based environment. The evolution of the ontology editors has to happen on several axes:

- From single-user applications to multi-user environments
- From thick client to thin clients (web-based)
- From single-user control of ontology content to multiple user content control

<sup>1</sup><http://protege.stanford.edu>

While the first two items can be generally solved by software re-engineering, managing the content of an ontology in a multiple user setting poses many conceptual challenges. When multiple users control and develop the content of an ontology, we need not only to resolve the technical issues of simultaneous access to the ontology for different clients, but also to address challenges specific to the collaboration aspect of the task: discussions, annotations, conflict resolution, and so on. We envision that ontology developers will use annotation to add comments to ontology components, to ask questions about them, to discuss modeling decisions and so on. This model raises the question on how the user annotations should be represented, stored and processed. The support for filtering and searching in the body of annotations are also essential features. A natural extension is to support the extraction of ontology views based on the user annotations. For example, a user may want to see a view of an ontology as agreed by most of the other users. This means that the tool should provide support for capturing user agreement or disagreement as well as ratings of other users annotations.

These are some of the research challenges that have to be tackled by a collaborative ontology development. In the following sections, we will present a prototype that we have developed, which supports some of the common tasks involved in collaborative ontology development.

## 2. COLLABORATIVE PROTÉGÉ SYSTEM

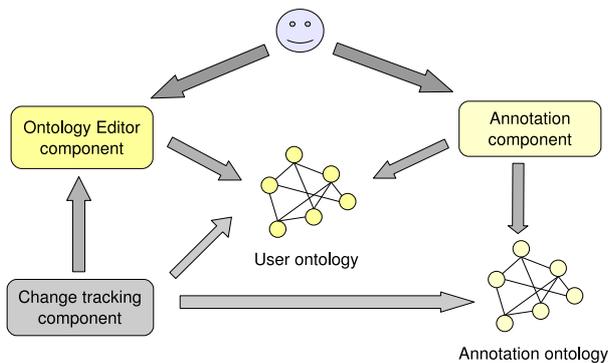
The Protégé system [1] is an open-source ontology editor and knowledge-base framework developed by Stanford Medical Informatics. It uses a frame-based representation formalism [3]. The Protégé editor also supports the editing of RDF(S)<sup>2</sup> and OWL<sup>3</sup> ontologies.

We developed the Collaborative Protégé prototype as an extension of the existing multi-user Protégé system. The multi-user system is a client-server version of Protégé, which allows multiple clients to connect to a Protégé server and to browse and edit simultaneously the same ontology. All the changes done by one client are seen immediately by all the other clients.

One of the main features of a collaborative development tool is to allow the users to comment and discuss about the content and changes of the ontology that they develop in common. We realize this feature by supporting the association of annotations to any component of the ontology or to

<sup>2</sup><http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>

<sup>3</sup><http://www.w3.org/2004/OWL/>



**Figure 1: Core components of the Collaborative Protégé architecture that support the annotation of ontology components and of changes in the ontology.**

any change that occurs in the ontology. The tool also provides support for different visualizations of the annotations, which can be customized by the user by specifying different filtering criteria. A user may search through the available annotations by specifying simple or complex search criteria.

The main purpose of this prototype is to gather users requirements for a collaborative ontology development environment. Currently, the prototype is available as a stand-alone application and as an applet. However, after gathering the users requirements and feedback, we envision that the functionalities offered by the current prototype will be implemented in a web-based application.

## 2.1 System Functionality

The system is an extension of the existing Protégé tool and takes advantage of the existing Protégé functionality, such as ontology editing and structured instance acquisition.

The main functionality provided by the Collaborative Protégé prototype are:

- Annotation of ontology elements, such as classes, properties, individuals
- Annotation of ontology changes, such as class creation, deletion, renaming, etc.
- Support for change proposals and voting of proposals
- Support for filtering of existing annotations
- Support for searching of annotations based on simple or complex criteria
- Support for discussion threads

Another important feature that is currently only experimental is the support of ratings for existing user annotations. This feature enables the implementation of different web-of-trust algorithms.

## 2.2 System Design

Figure 1 shows a diagram of the core components of the system that support the collaborative development of ontologies and specifically the ontology and changes annotation process. The user interacts with the *Ontology Editor Component* and the *Annotation Component*. The editing

component is provided by the underlying Protégé system. The *Annotation Component* allows the user to annotate ontology components, such as classes, properties and individuals, as well as ontology changes, such as class creation or deletion, with annotation types defined in the *Annotation ontology*.

The *Annotation ontology* is a RDF(S) ontology that provides the structure for the annotation types supported by the tool. The annotation types are extensions of the Annotea [2] annotations and contain concepts such as Comment, Advice, Example, etc. User annotations are stored as instances of the predefined annotation classes and can be used for annotating both ontology components as well as ontology changes.

The *Change tracking component* is responsible for intercepting the user actions in the GUI and creating change annotations attached to the changed ontology components. Change annotation types are defined in the *Annotation ontology*.

The system supports also discussion threads by allowing the users to reply to the comments of other users. This is realized by a flexible representation of annotations, which can themselves be annotated. The system supports also the rating and voting of proposals, which are represented as annotation types in the *Annotation ontology*.

Other components of the system are the *searching* and the *filtering* components, which are crucial in dealing with large bodies of annotations. The filtering component will be used in future versions of the system to create user-defined views of an ontology based on user preferences.

## 2.3 Software Architecture

The system is implemented as a plug-in of the existing Protégé system. The prototype enables several Protégé clients to connect through Remote Method Invocation (RMI) to a server that stores a repository of ontologies.

The current prototype can be used both as a stand-alone application for the cases in which several users edit the same ontology at different timepoints, or in a multi-user setting, in which multiple clients may edit the same ontology in a concurrent fashion. In the multi-user setting, the clients may connect either by using Protégé rich client or by using the Protégé applet client.

## 2.4 The Graphical User Interface

The graphical user interface of the Collaborative Protégé builds on top of the available plug-ins of the Protégé user interface and provides additional graphical components for editing and browsing the annotations. Figure 2 shows a screenshot of the Collaborative Protégé in which the classes view has been enhanced with an annotations panel in which the user can edit and browse the annotations.

The annotations panel provides different views of the annotations:

- Changes annotations view
- Ontology Components annotations view
- All annotations view (both Changes and Ontology Components annotations)
- Discussions thread view

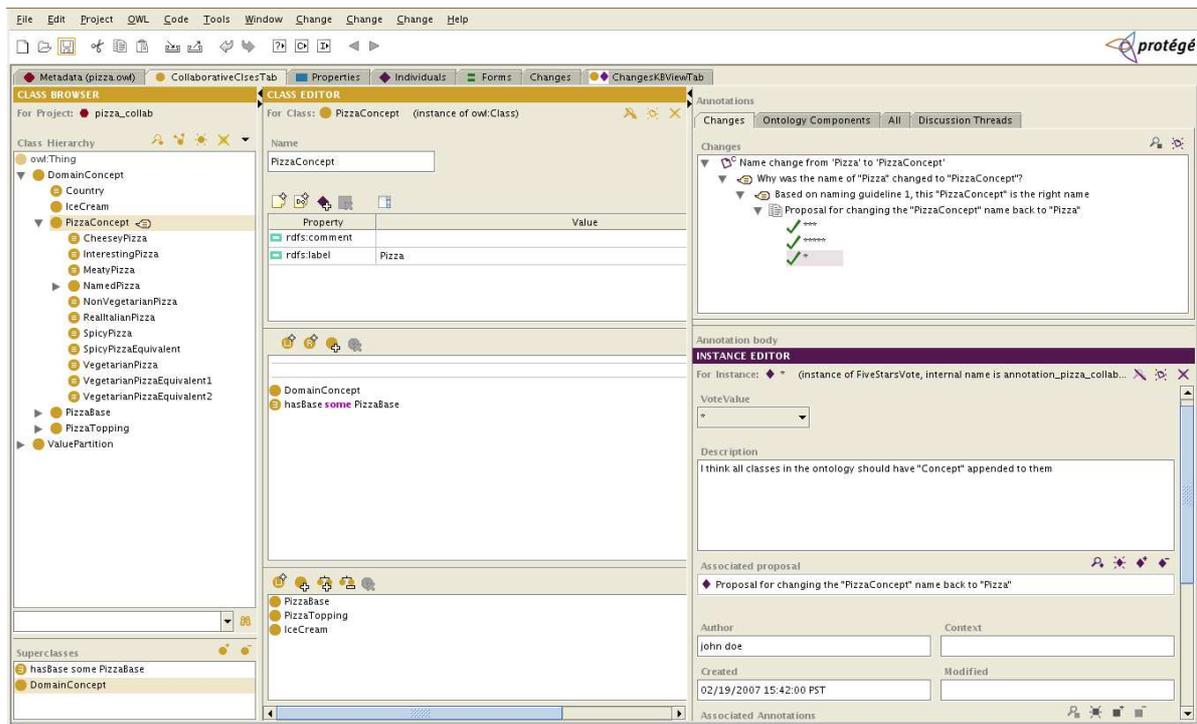


Figure 2: The Collaborative Protégé GUI. Classes view together with the annotations panel (on the right-hand side of the display).

The annotations tree that is shown in any of the views of the annotations panel can be filtered based on user preferences. For example, the user may choose to see only the annotations provided by a certain person, or the annotations of a certain type (for instance, only Example annotations), or annotations that have been added in a certain period of time. The user can also build complex filters that combine the criteria mentioned before.

A user may start a proposal for a change in the ontology and may start also a voting process for it. Two types of voting annotation types can be chosen: a “5-star” voting as shown in Figure 2, or a “Agree/Disagree” type of voting. Currently the prototype does not support a workflow for the voting mechanism, but it is one feature that we intend to support in the near future.

### 3. CONCLUSIONS AND FUTURE WORK

We have presented a prototype for collaborative ontology development that can be used for editing and browsing the user annotations on ontology components and on ontology changes. The prototype supports the searching and filtering of user annotations based on different criteria. Two types of voting mechanisms have been implemented that can be used in voting for change proposals.

The main purpose of the current prototype is to gather the user requirements for a collaborative ontology development environment. We envision that the existing functionalities and as well as new ones that result from the user feedback will be integrated in a web-based application with full ontology editing capabilities.

The existing Annotation ontology may be extended based on the user feedback. We also envision to provide workflow

support for different types of collaborative editing settings. For example, a possible collaborative editing scenario is in a community in which users do not have direct write permission on an ontology and a central authority reviews and accepts change proposals requested by the users. For this case, the system should provide support for the proposal and voting process.

We also intend to provide statistics of the annotations (for example, what ontology components have the most annotations associated to them) as well as a pluggable mechanism for supporting different web-of-trust algorithms.

### 4. REFERENCES

- [1] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of Protégé: An environment for knowledge-based systems development. Technical Report SMI-2002-0943, Stanford Medical Institute, 2002.
- [2] J. Kahan and M.-R. Koivunen. Annotea: an open RDF infrastructure for shared web annotations. In *Proceedings of the 10th International World Wide Web Conference*, pages 623–632, 2001.
- [3] N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of Protégé-2000: Combining interoperability and flexibility. In R. Dieng and O. Corby, editors, *Knowledge Acquisition, Modeling and Management, 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2-6, 2000, Proceedings*, volume 1937 of *Lecture Notes in Computer Science*, pages 17–32. Springer-Verlag, 2000.