Collaborative Ontology Development with Protégé

Tania Tudorache, Natasha Noy
Jennifer Vendetti, Timothy Redmond

Stanford Center for Biomedical Informatics

Tutorial at the Protégé conference 2009

Amsterdam, June 23, 2009
Outline

- Collaboration
- Multi-user Protégé – simultaneous editing
- Collaborative Protégé – collaboration support
- WebProtégé – web-based browsing and editing
- Moving forward
Outline

- **Collaboration**
- Multi-user Protégé – simultaneous editing
- Collaborative Protégé – collaboration support
- WebProtégé – web-based browsing and editing
- Moving forward
Collaborative Ontology Development

- **Collaboration**: several users contribute to the development of one ontology
  - Mechanisms for carrying out discussions, and
  - Reach consensus (e.g., polling, moderators)
  - Collaboration workflows – no one-size fits all
- Social platform for developing ontologies
- Each community does it its own way
Evolution of ontologies

- More prevalent in several domains
- “More” complex (taxonomies, classification → more formal representations)
- Larger
- More inter-dependencies (imports)
- Developed by domain experts rather than by ontology engineers
Examples of collaborative development

- Gene Ontology (GO)
- NCI Thesaurus
- BiomedGT
- OBI, BIRNLex, RadLex
- Open Biomedical Ontologies (OBO)
- International Classification of Diseases (ICD-10++)
How do groups collaborate?

- Send (tons of) emails!
- Telecons ...
- Use source control frameworks (sourceforge, Gforge)
- Use Web-based collaboration frameworks (WebEx, Centra)
- Google Groups, Docs, Sites (Wave – to come)
- … all is good, but how do I link all this to my ontology?!
What features would they need to support their collaboration?

- **Threaded discussions** in the context of an ontology:
  - Discuss about a class in the ontology
  - Discuss about a particular value of a property at a class/property/individual, or about an axiom (E.g., “I do not agree to the statement that Disease has synonym Disorder”)

- **Provenance information** (John did not agree last week, but he changed his mind this week)
What features would they need to support their collaboration? (cont.)

- **Complete change history**
  - *Bob created 3 classes yesterday, but today he was kind of lazy. He also tried to delete a class created by John but was not allowed to*

- Yes, your boss could make **statistics** … :)

- **Discuss** controversial changes

- **Undo** a change (quite a dangerous operation)

- Go back to any **previous state** in time of the ontology
  - *(Show me the ontology as it was yesterday at 4pm)*
What features would they need to support their collaboration? (cont.)

- **Personalized views** of an ontology based on:
  - User's role and tasks
    - *Reviewers of the Gene domain, will see only the Gene class subtree*
  - User's expertise
    - *Certain features are hidden for beginner users*
  - User's trust network
    - *Have the same ontology view as the other users in my group*
What features would they need to support their collaboration? (cont.)

- **Sandbox** capability
  - Try out some changes before others can see it
- **Versioning**
  - CVS and SVN – like, but also
  - Simultaneous editing
What features would they need to support their collaboration? (cont.)

- **Access policies**
  - User with different access rights: E.g., Editors are allowed to create content, while Reviewers are only allowed to read the ontology
  - “Classical” access rights: read, write
  - Different access right granularities:
    - Per ontology
    - Per part of ontology
What features would they need to support their collaboration? (cont.)

- Scalability, performance and reliability
- Flexible Workflow Support:
  - No two workflows are the same
  - Workflows evolve over time (e.g., requirements change)
  - Well thought-out workflows are needed: e.g., How should editing conflicts be handled?
  - Workflows and business rules should be enforced in the ontology tool
What they could use...

- Use ontology tools that support collaboration:
  - Collaborative Protégé
  - WebProtégé
  - BioPortal
- Other tools are available (see CKC 2007 workshop paper), but we are going to talk about x.Protégé
Outline

- Collaboration
- **Multi-user Protégé** – simultaneous editing
- Collaborative Protégé – collaboration support
- WebProtégé – web-based browsing and editing
- Moving forward
Multi-user Protégé
(a.k.a. Client - Server Protégé)

Ontology repository

- NCI Thesaurus (DB)
- pizza.owl
- newspaper.pprj

---

protégé

protégé

protégé

protégé

protégé
Multi-user Protégé - features

- Client-server architecture
- Simultaneous editing:
  - All changes are seen immediately by all clients
- Works with all backends of Protégé (OWL, frames, RDF)
- Recommended to be used with the database backend – support for transaction control
Starting the Protégé server

Execute the `run_protege_server` script from the Protégé installation directory

Server has started successfully and client can connect to it
Problems you may have..

- Server and client are behind a firewall
  - Solution: open 2 ports to be used by the server, and uncomment the ports line in the run_protege_script
  - Solution is described in detail on the Protégé wiki in the Multi-user tutorial

PORTOPTS="-Dprotege.rmi.server.port=5200 -Dprotege.rmi.registry.port=5100"
Configuring the server –
The Metaproject

- The metaproject is used to configure the server: users, access policies, projects
- Stored by default in Protégé installation directory
  /examples/server
Configuration of the Collaborative Protégé Server: The Metaproject

Projects available on the server

Policies on operations/project

Associated annotation project

Project location on server
The Metaproject - Policies

- Policies are operations allowed for a group of users: E.g. “World can Read, Write” and are attached to a project
- A policy can be reused for several projects
- Predefined operations (read, write, display in project list, admin, etc.) - see wiki
- Most predefined policies are enforced: Read, Write, etc.
- You can define your own policies and enforce them in your application using Protégé policy manager
Connecting to a Collaborative Protégé server

File menu -> Open ... -> Server

Web-start and applet demo: http://smi-protege.stanford.edu/collab-protege/
Selecting a remote project

Select from the projects list: **Collaborative Pizza**
After connecting this is what you should see..
Administering the Protégé server

- Can be done with the Admin application
- Allows you to see the current projects, their status, connected sessions, and to administer them; and shut down the server
- Needs special policy configured in the metaproject
The Server Admin - Projects

![Administer Protege Server (Logged in as Admin) window]

Remote projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Status</th>
<th>Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative Newspaper</td>
<td>SHUTTING_DOWN</td>
<td>[]</td>
</tr>
<tr>
<td>Collaborative Newspaper Annotations</td>
<td>READY</td>
<td>[]</td>
</tr>
<tr>
<td>Collaborative Pizza</td>
<td>READY</td>
<td>[Session(id=102, user=Guest), Session(id=106, user=Ray Fergerson)]</td>
</tr>
<tr>
<td>Collaborative Pizza Annotations</td>
<td>READY</td>
<td>[Session(id=107, user=Ray Fergerson), Session(id=103, user=Guest)]</td>
</tr>
<tr>
<td>Pizza</td>
<td>CLOSED_FOR_MAINTENANCE</td>
<td>[]</td>
</tr>
</tbody>
</table>

Details on project: Collaborative Pizza

Statistics
- Estimated round trip time (ms):
- 6
- Milliseconds to calculate frame cache:
- 6
- Transaction Isolation Level:
- NONE

Users

<table>
<thead>
<tr>
<th>Session ID</th>
<th>User</th>
<th>IP Address</th>
<th>In Transaction?</th>
<th>Server Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>Guest</td>
<td>127.0.1.1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>106</td>
<td>Ray Fergerson</td>
<td>127.0.1.1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Projects status

- A project on the server can be in 3 different status:
  - READY – users can open it
  - SHUTTING_DOWN – a shut down was scheduled, project is not available anymore
  - CLOSED_FOR_MAINTENANCE – project is not available, maintenance can be performed on it on the server side (and after that it can be restarted)
Shutting down a project for maintenance
Shutting down a project for maintenance

You may also cancel a shut down of a project. Clients will be notified that the project is going to be shut down with repeating notifications. One the project has shut down, the clients will close the connection to the server cleanly.
If you have the right privileges, you can kill a user session.
Outline

- Collaboration
- Multi-user Protégé – simultaneous editing
- Collaborative Protégé – collaboration support
- WebProtégé – web-based browsing and editing
- Moving forward
Collaborative Protégé Features

- Extension of existing Protégé system
- Support for:
  - annotating ontology components and changes in the ontology
  - discussion threads
  - proposals and voting
  - searching and filtering
  - defining users, groups, policies
- Works in Protégé 3.x OWL and Frames
- Available in multi-user and stand-alone modes
- Distributed with Protégé installation

http://protege.stanford.edu/doc/collab-protege/
Collaborative Protégé GUI

- **Collaborative Tabs**
- **Annotations**
- **Annotation details**

Annotations on DomainConcept:

- *ttania (10/24/08 07:59): This is a comment on the DomainConcept class*
- *ttania (10/24/08 08:02) Re: This is a comment on the DomainConcept class*

**Details**

- **Author**: ttania
- **Created**: 10/23/2000 21:59:10 GMT-06:00

**Subject**: This is a comment on the DomainConcept class

**Description**

This is an example of a comment on a class.

You may also insert internal links to other entities in the ontology. For example, if you refer to class Pizza, you select from the "Add Internal Link To" combo box, Class and select the class Pizza. This will generate the following link:

@http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl#Pizza'

If you click on the link above, a window will pop up with the details of class Pizza.
Basic collaborative mechanism in Protégé

- **Ontology editor component:**
  - basic ontology editing functionalities

- **Annotation component:**
  - user ontology is annotated with annotation instances from the Annotation ontology

- **Change tracking component:**
  - changes are stored as instance of the Annotation ontology
Ontologies for supporting the collaborative development process

- We used ontologies for representing:
  - **Ontology components** (e.g., classes, properties, individuals)
  - **Changes** (e.g., Domain changed for a property)
  - **Roles** (e.g., Manager, Editor, SME, etc.)
  - **Annotations** (e.g., comments on classes, or changes)
The Changes & Annotation Ontology (ChAO)
The Collaboration Framework

Ontologies supporting the collaboration process

API access

Ontology repository

- Changes API
- Annotations API
- Workflow API
- Ontology Access API
- Policy Manager

Ontologies supporting the collaboration process:

- Discussions and Proposals
- Changes
- Ontology components/axioms
- Workflow (activities, tasks, control structures)
- Roles
- NCI Thesaurus
- Guidelines Ontology
- Gene Ontology
- ...
The Collaborative Framework + Clients

Collaborative Protégé

WebProtégé

Collaborative Framework

Ontologies supporting the collaboration process

API access

Ontology repository

Changes API
Annotations API
Workflow API
... Policy Manager

Ontology Access API

NCI Thesaurus
Guidelines Ontology
Gene Ontology
RadLex
Annotations and Discussion Threads

- Annotations are linked to a specific ontology component
- Different types of annotations
- Annotations types can be extended with no extra coding
- Users may annotate:
  - classes
  - properties
  - individuals
  - the ontology as a whole
- Annotations may be filtered and searched based on different criteria
Changes Tab

- See the history of a concept
- Users may comment on changes; for example on a class rename operation or on a change of a domain property
- Browse the change details (e.g. author, creation date, sub-changes, etc.)
Chat Tab

- Exchange live messages between users connected to the same Protégé server
- Supports HTML formatting (hyperlinks, bold, italics, etc.)
- Internal links to ontology entities (e.g., @'Gene')
- Chat available also as a Tab plug-in
Outline

- Collaboration
- Multi-user Protégé – simultaneous editing
- Collaborative Protégé – collaboration support
- WebProtégé – web-based browsing and editing
- Moving forward
WebProtégé – an alternative client for Collaborative Protégé

- WebProtégé is an open source light-weight ontology editor for the Web
- It is a Collaborative Protégé client
- Main features of WebProtégé:
  - Browsing of ontologies on the Web
  - Lightweight ontology editing
  - Designed using a portal metaphor: The user interface is composed of reusable components, called portlets
  - Customizable UI by drag-n-drop and by showing or hiding different ontology tabs
  - Extensible: Developers may easily implement their own tabs and portlets
GWT – Google Web Toolkit

- GWT: write web client applications in Java
- GWT compiles the Java code into optimized JavaScript
- The server side is written in Java
- Client-Server communication through RPC (JSON, others also available)
WebProtégé - Architecture

- 2 sides of the story: server and client
- Server is completely implemented in Java and makes API calls to Protégé
- Client side is developed in Java, and later compiled into JavaScript
- Communication between server – client is done with GWT RPC (can be done in other ways, too)
WebProtégé UI

- Portal metaphor (inspired from MyYahoo and iGoogle)
  - Portlets are panels that have some functionality (e.g. display the properties of a class) and can be dragged around
  - Users can customize the layout of WebProtege:
    - Enable/disable tabs
    - Customize the portlets in a tab
    - Configuration is stored on server and it per project/user
- Uses a 3rd party widget library: GWT-ext
WebProtégé GUI – showing the NCI Thesaurus

http://bmir-protege-dev1.stanford.edu/webprotege/
First page in WebProtége

Select a project

Open the same ontology in Collaborative Protége for editing

http://bmir-protege-dev1.stanford.edu/webprotege/
Opening Collaborative Pizza

Login to edit

Discussions

Drag-n-drop, resize, close portlets
WebProtégé installation

- Tomcat or another servlet engine
- Download the war file from the WebProtégé wiki page
- Follow the step-by-step Administrators' Guide
Extending WebProtégé

- Plug-in infrastructure very similar to Protégé's: create your own tabs and portlets
- Just extend: AbstractTab and AbstractEntityPortlet
- Implement your own RPCs, if needed
- Reuse existingportlet code
- Writing a tab – as easy as creating an empty class that extends AbstractTab
- Promise to have soon some more documentation :)
Outline

- Collaboration
- Multi-user Protégé – simultaneous editing
- Collaborative Protégé – collaboration support
- WebProtégé – web-based browsing and editing
- Moving forward
What's next?

- We are far from done …
- Near future:
  - Migrate Collaborative Protégé and WebProtégé to Protégé 4
  - WebProtégé:
    - Finalize the plug-in mechanism
    - Improve the editing support
    - Migrate to a new widget library
    - Documentation!
What's next? (cont.)

• Medium term:
  – Support collaboration features that are not yet available
  – Conflict management
  – Workflow support
  – Evaluate the tools in different projects (We need your help!)

• Long term:
  – Who knows :)

Get involved!

- If you would like to get involved, talk to us!
- Contributions from the community are welcome
- Implement your own plug-ins for WebProtégé
- If you used any of the tools, we would be happy to get your feedback!

THANK YOU!
Resources

- Collaborative Protégé Users' Guide

- Collaborative Protégé paper:

- WebProtégé short paper:

- Try WebProtégé out:

- Accessing Collaboration features from other applications:
  - http://protegewiki.stanford.edu/index.php/ChAO_API

- Protégé client-server tutorial