

WebProtégé 3.0 – Collaborative OWL Ontology Engineering in the Cloud*

Matthew Horridge, Rafael S. Gonçalves, Csongor Nyulas,
Tania Tudorache, and Mark A. Musen

Stanford Center for Biomedical Informatics Research

1 Introduction

In this demonstration, we present the main features of WebProtégé, a free, open-source, cloud-based OWL ontology editor that supports simultaneous collaborative editing for geographically distributed users. WebProtégé is available for general purpose use at <https://webprotege.stanford.edu>, where it hosts over **55,000 OWL ontology projects** and over **42,500 user accounts**. The software is open-source and all source code is available on GitHub¹.

WebProtégé 3.0 builds on the success of previous versions but features a new modern interface and new features for collaboration and review. This demonstration, at the ISWC2018 posters and demos session will enable attendees to see the latest features as well as providing them with an opportunity to interact with, and provide feedback to, the WebProtégé developers. We discuss the main features here.

2 Main Features

WebProtégé is essentially a “Google docs” for editing ontologies. Users access “projects” that are collections of OWL ontologies augmented with a change history, and issues/comments. WebProtégé provides a default user-interface that supports lightweight ontology editing, for ontologies that more or less fit within the OWL2EL profile. However, it can also be configured for editing ontologies that require full-blown OWL 2 axioms and class expressions.

Ontology Editing Figure 1 shows the main editing interface, which provides the familiar tabbed look and feel that is common to all desktop versions of Protégé. The classes tab is shown here, which comprises four resizable views: (Class Hierarchy, Class Description, Comments and Project Feed). Users can add, remove and customize the layout of views on a tab and the state of the user-interface is automatically persisted for each user in the context of a project.

In contrast to previous versions, WebProtégé 3.0 uses proper URL routing, meaning that entities, such as classes, properties and individuals, and the tabs

* Supported by Grant GM121724 from the National Institute of General Medical Sciences at the United States National Institutes of Health.

¹ <https://github.com/protegeproject/webprotege>

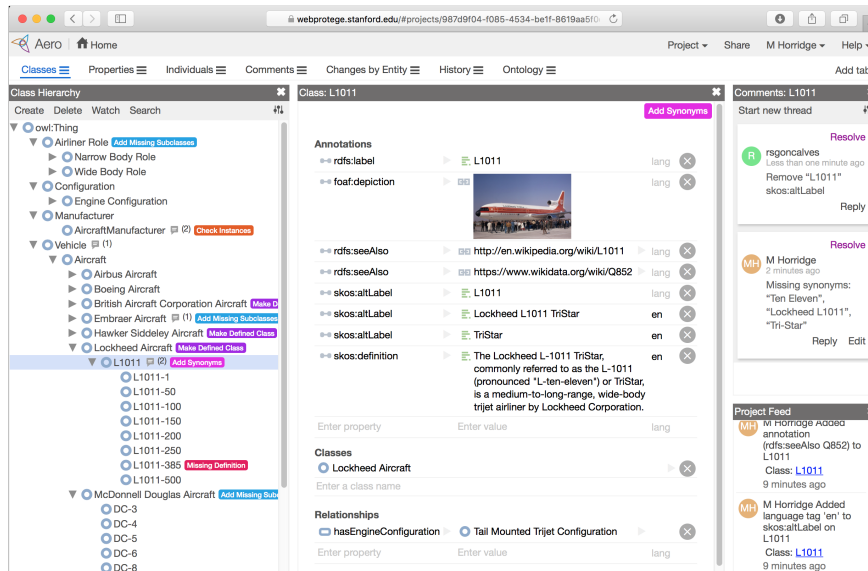


Fig. 1. Class editing in WebProtégé. This figure demonstrates the default user interface and shows the use of tags, comments/issues and the project feed.

(or pages) that display them, are bookmark-able links that can be shared with people.

WebProtégé allows ontology edits to be seen by all users as they take place. However, it is possible to set up a project so that users may only view or comment on a project for cases where there is a core group of editors and a larger group of commenters/reviewers.

Collaborative Features Besides collaborative ontology editing functionality, WebProtégé also allows users to comment on a project and create issues in the form of *threaded comments*. Comments sit outside of an ontology but pertain to specific entities in an ontology. They are formatted in Markdown and can contain links to any entities contained in the signature of the ontology as well as GitHub style @user mentions. Discussion threads of comments for a given entity can be seen on the right-hand side of Figure 1 in the Comments view. The number of threads is also displayed next to entity names in the various entity hierarchies (left hand side of Figure 1). Threads can be created and closed as issues are opened and then dealt with by ontology editors. Figure 2 shows a global view of the comments manager that is available in the default interface.

When comments are posted, project participants are notified via email (left inset in Figure 2). Emails contain the body of the comment and clickable links that take user directly to the comment in a browser window. WebProtégé also provides integration with *Slack*² and makes it possible to associate a *Slack* Web-

² <https://slack.com>

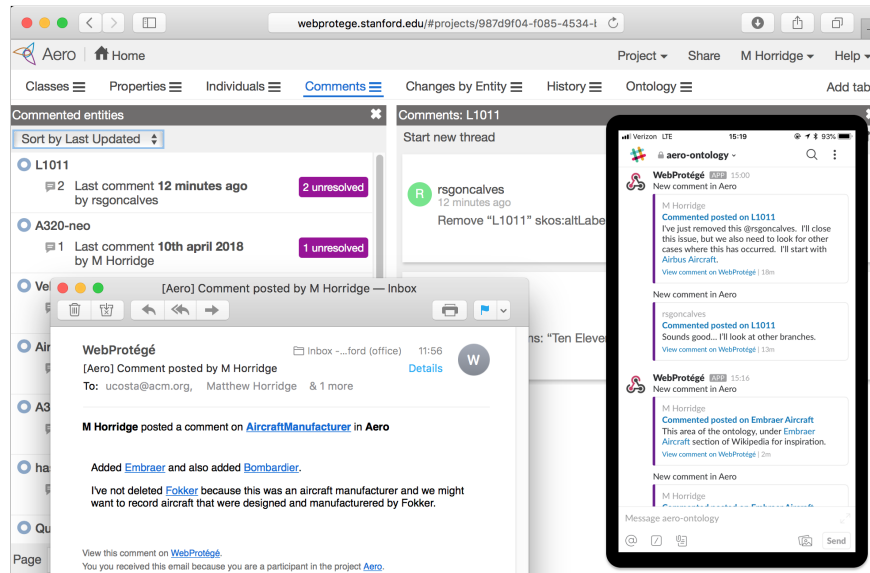


Fig. 2. Comments and Issues in WebProtégé. The main part of the figure shows the comments manager view. Comments can be sorted and viewed by entity. The lower left-hand side inset shows a notification email sent out to participants after a comment has been posted. The right-hand side inset shows integration with the chat app Slack.

hook³ with a project so that comments are posted to a Slack channel for external notification. Slack users are able to click on links in their Slack clients in order to view the comment directly in WebProtégé (right inset in Figure 2).

In addition to comment threads, which can be used for adding review comments to a project, WebProtégé also allows entities to be “tagged” with badges that can be defined for a project. Figure 1 shows various tags in the class hierarchy that have been added as part of the review process. Tags and comments usually go hand-in-hand. Entities with specific tags can also be searched for in the search dialog.

Change Tracking and Project History One of the key features of WebProtégé is that all ontology changes are tracked and grouped into revisions. Revisions are based on atomic user-interface operations and it is possible to revert a single revision. Figure 3 shows an example of the history of a project. Changes can also be viewed/filtered by entity, so that the entire history for an entity description (both logical and non-logical) can be examined. Labels for revisions are automatically generated based on the ontology changes in a revision. In the full-blown OWL 2 editing interface (not shown here) it is possible to group a set of changes into a commit, which can then have a commit message assigned to it. It is also possible to download the set of ontologies in a given revision.

³ <https://api.slack.com/incoming-webhooks>

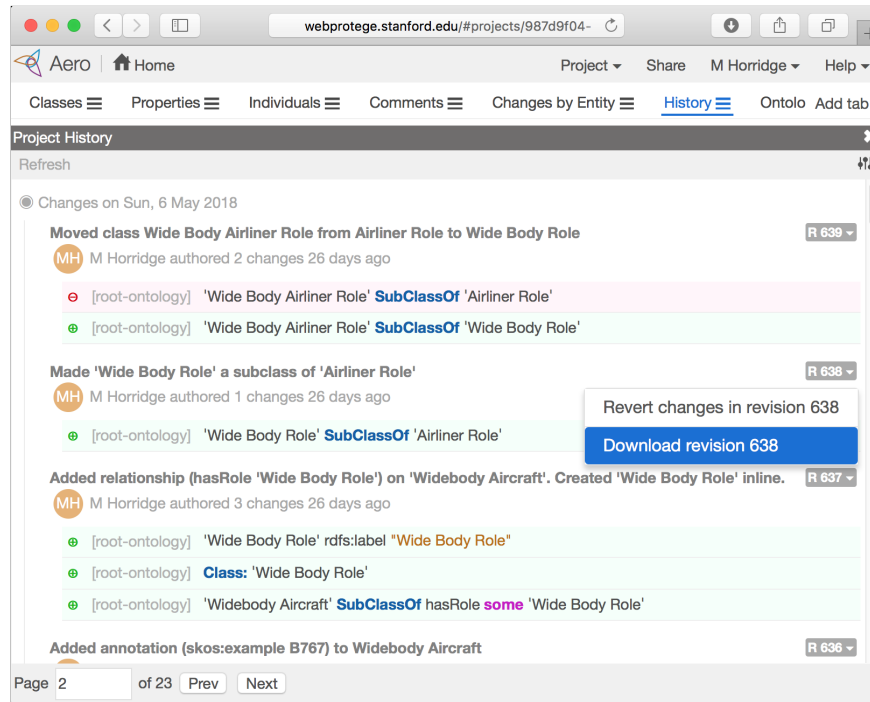


Fig. 3. Change Tracking in WebProtégé. All ontology changes are tracked and grouped into revisions that can be reverted and downloaded. Revisions are labelled with a description, author and timestamp.

Integration with Third Party Applications WebProtégé does not have a traditional “drop-in” plugin mechanism like the desktop version of Protégé does. However, going forward, we intend to facilitate integration with third-party applications via Webhooks and a Web-API. As of now, third-party applications can be notified of the changes to a project by adding an event Webhook to a project. JSON formatted messages are posted to the URL associated with the Webhook when changes in the associated project occur. We expect that this functionality will be used for loose coupling with third-party applications in the same way that GitHub and Slack can integrate with other applications.

3 Summary

WebProtégé is developed by the Protégé group at Stanford Center for Biomedical Informatics Research. It is under active development and we welcome feedback from the Semantic Web community in order to guide future development directions. This work is supported by Grant GM121724 from the National Institute of General Medical Sciences at the United States National Institutes of Health.