

Protégé (<http://protege.stanford.edu>) is an open-source tool for editing and managing ontologies. It is the most widely used domain-independent, freely available, platform-independent technology for developing and managing terminologies, ontologies, and knowledge bases in a broad range of application domains. Protégé is implemented in Java, and runs on a broad range of hardware platforms, including Windows, MacOS, Linux, and Unix. The community of Protégé users exceeds 25,000, and to date there have been seven annual international Protégé conferences, with over 220 attendees drawn to the latest meeting. The large and active Protégé user community regularly contributes enhancements to Protégé software and participates in online discussion groups devoted to modeling questions, technical-support issues, and requests for new features. Protégé's core functionality can be extended in many ways by creating plug-ins. There are over 90 Protégé plug-ins providing advanced capabilities such as reasoning and inference support and visualization of large ontologies.

Protégé has been used as the primary development environment for many ontologies in the life sciences. These projects include the Foundational Model of Anatomy, Cerner's Clinical Bioinformatics ontology, the DICE TS, and the MGED Ontology. Protégé has become the most widely used tool in the world for use with the Ontology Web Language (OWL); as more and more designers of biomedical ontologies adopt this W3C recommendation, Protégé is becoming the ontology-editing environment of choice for these developers.

The basic Protégé infrastructure provides the following key features which makes it particularly suitable to the needs of developing and using biomedical ontologies in a variety of applications:

- **An extensible knowledge model.** The internal representational primitives in Protégé can be redefined declaratively, permitting Protégé to have representations appropriate for a variety of ontology languages. Protégé's representational primitives—the elements of its knowledge model—provide classes, instances of these classes, slot representing attributes of classes and instances, and facets expressing additional information about slots.
- **A customizable user interface.** The standard Protégé user interface components for displaying and acquiring data can be replaced with new components that fit particular types of ontologies best (e.g., for OWL).
- **Ability to import ontologies in different formats.** There are several plug-ins available for importing ontologies in different formats into Protégé, including DAG-EDIT, XML, RDF, and OWL.
- **Support for data entry.** Protégé provides facilities whereby the system can automatically generate data entry forms for acquiring instances of the concepts defined by the source ontology.
- **Ontology authoring and management tools.** The PROMPT tools are plug-ins to Protégé that allow developers to merge ontologies, to track changes in ontologies over time, and to create views of ontologies. The Protégé internal knowledge representation can be translated into the various representations used in the different biomedical ontologies. Protégé has different back end storage mechanisms, including relational database, XML, and flat file.
- **An extensible architecture that enables integration with other applications.** Protégé can be connected directly to external programs in order to use its ontologies in intelligent applications, such as reasoning and classification services.
- **A Java Application Programming Interface (API).** System developers can use the Protégé API to access and programmatically manipulate Protégé ontologies.

Protégé can be run as a stand-alone application or through a Protégé client in communication with a remote server. We have also developed a Protégé Web browser, which allows users to share, browse, and edit their ontologies using a standard Web browser. In addition, users can create and post comments about specific ontology components. Thus, the browser also provides a rudimentary Web-based collaboration environment that can help communities in developing biomedical ontologies.

We have also developed a tool for comparing ontology versions. PromptDiff automatically calculates a structural difference between ontology versions, identifying which concepts have changed between versions. It identifies both simple and complex changes (such as moving classes, or adding or deleting a tree of classes, and modifying properties), and it presents the comparison results to the user in an intuitive way. Users then can accept or reject the changes between versions.

Jambalaya is a Protégé plug-in that provides an extensible, flexible, and scalable visualization environment for exploring, navigating, and understanding ontologies. Classes and instances are represented as nodes in a graph; different types may be distinguished using different color hues. Directed edges (arcs) are used to show relationships between concepts and instances.

In conclusion, Protégé can be a useful tool for managing and using biomedical ontologies. It can access the breadth of bio-ontology content in a single tool. Its graphical user interface components can help researchers and developers visualize and browse large and complex ontologies. Protégé can help them align and compare ontologies that overlap in content as well as different versions of ontologies. The Protégé platform supports Web-based ontology viewing and collaboration and it provides different back-end storage mechanisms. We believe that Protégé can help the biomedical community leverage the diversity of existing knowledge sources and create more robust and useful ontologies.