Simplifying MIREOT: a MIREOT Protégé Plugin

Josh Hanna[1,2], Cheng Chen[2], W. Alex Crow[2], Roger Hall[2], Jie Liu[2], Tejaswini Pendurthi[2], Trent Schmidt[2], Steven F. Jennings[2], Mathias Brochhausen[1], William Hogan[1]

[1] University of Arkansas for Medical Sciences
jhanna@uams.edu
[2] University of Arkansas at Little Rock

Abstract. The Web Ontology Language (OWL) [1] is a commonly used standard for creating ontology artifacts. However, its capabilities for reusing existing OWL artifacts in the creation of new artifacts is limited to the import of whole ontologies, even when only a small handful of classes, object properties, and so on (which we refer to generically as "OWL components") are relevant. This situation can result in extremely large and unwieldy, or even broken, ontologies.

To address this problem while still promoting ontology reuse, the OBI Consortium [2] has elucidated the Minimum Information to Reference an External Ontology Term (MIREOT) [3]. We provide a suite of plugins to the Protégé [4] editor that greatly simplifies the use of MIREOT principles during ontology creation and editing.

1 Introduction

Dealing with the ever-increasing amount of data being generated has been an extremely difficult problem. Researchers have spent a lot of time categorizing this information and the semantic web is one response to this problem. Ontologies, in particular, are one of the technologies that the semantic web uses to help organize information. To make ontologies the basis of semantic integration across multiple data sources, we need to facilitate re-use of ontologies – and representational units from those ontologies – across communities.

However, the semantic-web standard for ontologies, the Web Ontology Language (OWL) [1], has limited built-in facilities for reuse. The only way an OWL-based ontology can re-use work done in other OWL ontologies is by importing the entire ontology. Given the considerable size of some ontologies (e.g. Chemical Entities of Biological Interest [5] and the Ontology of Biomedical Investigations [2]), this can be a major obstacle to ontology re-use among ontology developers.

For this reason, the Ontology for Biomedical Investigations (OBI) Consortium [2] has defined the Minimum Information to Reference an External Ontology Term (MIREOT) [3] to facilitate sharing of components of OWL ontologies such as individual classes rather than entire imports of ontologies. Xiang et al [6] describe an implementation of the MIREOT principles in OntoFox, a web service for creating subsets of the classes and object properties of a source ontology for import into an ontology under development. However, for users of ontology editors like Protégé [4], the fact that OntoFox is a web service makes its integration into the ontology development process somewhat unwieldy. The user must go outside Protégé, generate import files using OntoFox, and then import those files into the ontology in Protégé.

While the specific issues we want to resolve are linked to the OWL language, our primary aim is to provide a simple MIREOT solution for the user community of Protégé. For that purpose, we present a cohesive suite of plugins built for Protégé 4.1 to simplify the use of basic MIREOT principles.

2 Background

The OWL language is designed to fascilitate data sharing on the web in a machine readable way through the creation of ontologies. All OWL ontologies can be expressed as an RDF [7] graph, an important type of datastore for the Semantic Web. The most current version as of this writing is OWL 2.

Protégé is a popular, open-source ontology editor built and maintained by researchers at Stanford. It provides a graphical interface that uses the OWL API [8] to build and maintain ontologies. It also includes an OSGI-compatible framework that makes it easy to extend via plugins.

A plugin [9] for Protégé 3 with similar functionality was developed by the Stanford Center for Biomedical Informatics Research. However, this plugin has not yet been updated for Protégé 4 and thus lacks support for OWL 2.

3 Methods

The OBI Consortium has defined a set of principles – which we refer to generically as "MIREOT principles" – for importing only one or more OWL components from another ontology. These principles specify the minimum unit of import as follows: the IRI of the source ontology, the IRI of the source component being imported (e.g., class, object property, etc.), and the IRI of the new parent component in the target ontology. Additionally, the Consortium defined a new annotation property in the Information Artifact Ontology (IAO)[10], named "iao:imported_from". The value of this property on the imported component is the IRI of the source ontology, ideally. Finally, they recommend that certain relevant annotations be imported, such as "rdfs:label" and "iao:definition".

We have implemented crucial parts of the MIREOT specification in a suite of Protégé plugins. Our implementation of the MIREOT specification differs in the following ways:

1. It pulls every annotation of the source component, not just "rdfs:label" and "iao:definition".
2. It saves all relevant axioms in the target ontology instead of external files.

Additionally, we made the following assumptions:

1. Only the most recent source ontology is preserved in the ”iao:imported_from” annotation property. Any similar annotations are stripped from the component to avoid confusion about which ontology was the direct source of a class.
2. If an ontology lacks an IRI, then the URL from where it was loaded is used instead.
3. If an annotation property that is not found in the target ontology is imported, then its “rdfs:label” is transferred as well, if it exists.

4 Results

The suite of plugins includes one tab plugin that contains three view plugins. A tab plugin tends to be simple; it contains multiple views that should work together to accomplish some overarching purpose. A view plugin represents the building blocks of a tab plugin and implement most of the actual functionality within a tab plugin. The tab plugin does not manage any additional state.

The first and most novel of the three view plugins is the “MIREOT Additional Ontology View”. It allows users to load an additional ontology and search it for classes or object properties. The search algorithm matches the query against a configurable set of annotations on each component. The currently searchable annotation properties are “rdfs:label”, “rdfs:comment”, and “iao:definition”. All loading and searching is done in a separate thread to keep the user interface fluid and stable.

The other two view plugins represent the active ontology loaded by Protégé. They display the hierarchy of either the classes or the object properties of the active ontology. Additionally, they allow a user to drag a component from the results panel of the “MIREOT Additional Ontology View” and drop it under the new parent of the component in the target ontology.

A short video demonstrating a subset of the features of the suite can be found online at http://vimeo.com/46860781. The suite and installation instructions can be found online at http://bitbucket.org/sohj/mireot-protege-plugin.

5 Discussion and Future Work

We have developed a suite of Protégé plugins that implements crucial parts of the MIREOT specification defined by the OBI Consortium. To our knowledge, this is the first Protégé plugin to do so. Although other tools such as OntoFox also implement MIREOT, our plugins place the workflow directly into the most commonly used ontology editor, and thus are more likely to facilitate MIREOT in practice. Although it has face validity, verification of this conjecture requires further study.

Our implementation of the MIREOT principles differ from the approach taken in the OBI specification in several notable ways:

- We import all of the annotations of an imported component. We do this because vital information about some components are included in annotations other than "rdfs:label" and "iao:definition".
- We keep the axioms in the main ontology. This decreases the friction of development for Protégé users while still maintaining necessary provenance information for editors, consumers, and users.
- We report only the most recent source ontology information. This will prevent confusion based on multiple "iao:imported_from" annotations that do not allow one to distinguish from which ontology a component was imported directly.
- We save the ontology URL as the IRI in the case of ontologies that do not have an IRI. We feel, it is important that the provenance of the imported component is preserved somehow.

One of the major reasons that OBI separates the imported terms from the main ontology is to allow for automated updates. For this reason, we intend to develop a tool which will allow users to update the terms while still keeping them in the main ontology using the saved provenance information.

6 Acknowledgements

This work was funded by award R01GM101151 from the National Institute for General Medical Sciences. This paper does not represent the views of NIGMS or the National Institutes of Health.

References

1. W3C Consortium, http://www.w3.org/TR/owl2-overview
2. OBI Consortium, http://purl.obolibrary.org/obo/obi
3. Courtot Ml, Gibson F, Lister AL, Malone J, Schober D, Brinkman RR, et al. MIREOT: The minimum information to reference an external ontology term. Applied Ontology. [10.3233/AO-2011-0087]. 2011;6(1):23-33.
4. **Protégé**, http://protege.stanford.edu
5. de Matos, P., Alcntara, R., Dekker, A., Ennis, M., Hastings, J., Haug, K., Spiteri, I., Turner, S., and Steinbeck, C. (2009). Chemical entities of biological interest: an update. Nucleic Acids Res. in the press.
6. Xiang Z, Courtot M, Brinkman RR, Ruttenberg A, He Y. OntoFox: web-based support for ontology reuse. BMC research notes. 2010;3:175.
7. W3C Consortium, http://www.w3.org/TR/rdf-concepts/
8. Matthew Horridge, Sean Bechhofer. The OWL API: A Java API for OWL Ontologies. Semantic Web Journal 2(1), Special Issue on Semantic Web Tools and Systems, pp. 11-21, 2011
9. Stanford Center for Biomedical Informatics Research, http://protegewiki.stanford.edu/wiki/BioPortal_Import_Plugin
10. OBI Consortium, http://purl.obolibrary.org/obo/iao.owl