Annotating OWL Ontologies

Bijan Parsia and Aditya Kalyanpur

University of Maryland, MIND Lab, 8400 Baltimore Ave,
College Park MD 20742, USA
{bparsia}@isr.umd.edu
{aditya}@cs.umd.edu

1 Introduction

One key function of the Semantic Web is to support knowledge representation based annotations of Web "resources". The linguistic apparatus (e.g., OWL[1]) used to express these annotations (e.g., OWL classes, or properties) are themselves Web resources and hence subject to annotation. To this end, we have integrated Annotea[2] based human and machine oriented annotations into our OWL ontology editor SWOOP.[3]

2 OWL Entities and Annotation Properties

OWL has four sorts of entity that are of primary interest to an annotator or ontology user: ontologies themselves, classes, properties, and individuals. In the DL species of OWL, the expressiveness of the language for describing each sort of entity varies greatly, from the wide range of constructors and sorts of axioms for defining classes to the minimal vocabulary (with minimal logical impact) for describing ontologies themselves. By contrast, the Full species of OWL allows for the entire language to be used for any sort of entity.

OWL DL does have one sort of assertion which can be uniformly applied to all sorts of OWL entity: owl:AnnotationProperty[4][5] based assertions. From the model theoretic point of view, such assertions are mere comments. All annotation properties are ignored by the reasoner, and they may not themselves be structured by further axioms.[6] owl:AnnotationProperty assertions can have as objects either individuals or datavalues, including rdf:XMLLiterals, thus can embed arbitrary XML, including RDF/XML (e.g., Annotea comments), XHTML, or SVG. The built in annotation properties rdfs:label and rdfs:comment are already extensively used in user interfaces (e.g., tool tips) and in other end user displays.

[1] The W3C's Web Ontology Language: http://www.w3.org/2001/sw/WebOnt/
[2] http://www.w3.org/2001/Annotea/
[3] http://www.mindswap.org/2004/SWOOP/
[4] In this paper, the 'rdf','rdfs', and 'owl' are mapped to the obvious namespaces.
[5] http://www.w3.org/TR/owl-ref/#Header
[6] OWL DL is probably more restricted than it needs to be. For example, some form of subproperty reasoning over annotation properties is quite feasible.

3 Annotea for OWL

The Annotea project developed an infrastructure for the creation and sharing of out of band, fine grained, extensible, Web based annotations. The Annotea framework has two parts: an RDF based annotation format and a protocol for publishing, retrieving, and managing those annotations. We have extended the Annotea format to support machine oriented annotations for collaborative editing. We have also investigated other distribution mechanisms such as OWL annotation properties and RSS 1.0 feeds.

We have written an Annotea client plugin for SWOOP. The SWOOP plugin can publish and display annotations using the standard Annotea vocabulary (including support for various annotation types such as questions, explanations, examples, etc.) to the Annotea server or to an annotation property in an OWL document.

We have also defined an OWL ontology for a new class of annotations — ontology changes. The Annotea project did define a "Change" class of annotations, but it is designed to indicate a *proposed* change to the annotated document, with the proposal described by a chunk of HTML-marked-up natural language. In our ontology, annotations correspond to specific *edits* (assertions, deletions, modifications) made in SWOOP and SWOOP can read and *apply* those edits .

SWOOP uses the OWL API[7] to model ontologies and their associated entities. The OWL API separates the representation of changes from the application of changes. Each possible change type has a corresponding Java class in the API, which are subsequently applied to the ontology (essentially, the Command design pattern). These classes allow for the rich representation of changes, including metadata about the changes. We have used these classes as the basis for our changes annotations. Using this change' ontology, SWOOP can externalize and export annotated change sets, which can then be browsed, filtered, endorsed, recommended, and selectively accepted. Thus, it is possible to define "virtual versions" of an ontology by specifying a base ontology and a set of changes to apply to it.[8]

4 Granularity

Annotea uses XPointer[9] to associate annotations with fine grained parts of documents. For classes, properties, and individuals, there isn't a lot of further granularity to be had. The URI of a class gets you that class and classes do not have subranges. Class *descriptions*, as collections of axioms, have an interesting and fine grain, while as collections of RDF *triples* they have an even finer grain, though of disputed use. We plan to explore XPointer schemes[10] which address the definitions. It is likely that we will focus above the triple layer, i.e., on axioms.

[7] http://owl.man.ac.uk/api.shtml
[8] This mechanism is modeled on Smalltalk's change records and sets.
[9] http://www.w3.org/TR/xptr/
[10] http://www.mindswap.org/papers/swrp-iswc04.pdf