A Tool for Working with Web Ontologies

Aditya Kalyanpur, University of Maryland, USA
Bijan Parsia, University of Maryland, USA
James Hendler, University of Maryland, USA

ABSTRACT

The task of building an open and scalable ontology browsing and editing tool based on OWL, the first standardized Web-oriented ontology language, requires the rethinking of critical user interface and ontological engineering issues. In this article, we describe Swoop, a browser and editor specifically tailored to OWL ontologies. Taking a “Web view” of things has proven quite instructive, and we discuss some insights into Web ontologies that we gained through our experience with Swoop, including issues related to the display, navigation, editing, and collaborative annotation of OWL ontological data.

Keywords: computer systems; Semantic Web; Web technologies

INTRODUCTION

The Web ontology language, OWL (Dean & Schreiber, 2004), was approved in February 2004 as a World Wide Web Consortium (W3C) Recommendation for the publication of ontologies on the World Wide Web—creating a standard language for the publication and exchange of ontological models on the Web. OWL reflects almost 10 years of research, experimentation, and small-scale deployment of Web ontologies; a number of certain features in its design were made explicitly to help realize the ideal of Web-based ontologies, that is, of integrating knowledge representation with the open, global, and distributed hypermedia system of the Web, compatible with the principles of Web architecture design. In this article we discuss some insights into supporting the use of Web ontologies that we have gained in building Swoop, an ontology browser and editor, designed specifically for use with OWL and directly supporting the use of Web-based “cultural metaphors”—that is, based on the way people are used to interacting with documents and data in current Web applications.
A WEB (ONTOLOGY) BROWSER—OWL

OWL is a standard for representing knowledge on the Web, with a focus on both making these documents compatible with Web standards and on being useful for the modeling of knowledge using past research on ontologies and reasoning. OWL comes in three increasingly expressive sublanguages—OWL Lite, DL, and Full. The Lite and DL species of OWL are based on description logics, that is, decidable, class- and property-oriented subsets of first-order logic. OWL Full follows RDF schema in having a higher-order syntax (although first-order semantics)—OWL Full does not enforce a strict separation of classes, properties, individuals, datatypes, or data values. Any entity could be, for example, both a class and an individual. This design was motivated by the Web architecture dictum that “everything is a resource,” thus an individual, and from the general modeling consideration that the choice between whether to represent some aspect of a domain as a class or an individual is not always clear. In a world where people are trying to reuse vocabulary and map between concepts, it seems quite natural to be able to express the dual view of certain domain objects as either classes or individuals, and sometimes both.

One characteristic of “Webized” languages, especially Semantic Web languages, is the systematic prevalence of Universal Resource Indicators (URIs)2 as names for most entities. In OWL, names for classes, properties, individuals, datatypes, and so forth are URIs. URIs have a number of useful properties, including:

1. For a number of URI schemes, notably http URIs, there is a well-developed set of mechanisms for avoiding name collisions, most notably the domain name system (DNS).
2. These mechanisms, especially the DNS, interact with various Internet protocols, notably HTTP, to make it very easy to publish and retrieve information associated with a URI.
3. URIs have various degrees of opacity. For example, HTTP imposes relatively few constraints on the semantics of the scheme specific part 1. A URI is a generalization of the more common URL, roughly composed of a naming scheme or protocol indicator (http, ftp, mailto, etc.), a unique indicator (a domain name space name for http, a mail address for mailto), and a “fragment id,” which is a hash mark followed by a set of characters—thus, for example, an OWL class called “person” from an ontology on a university server might be named by the URI: http://www.thisuniversity.edu/OntologyLib/csontology#person. The hierarchical structure seen in most http URIs can map directly into a file system (which is a very useful default behavior), but it can also map into queries on a relational database, the object structure of a long-running process, or any other Web resource.
4. URIs can work well for end users, who have developed a lot of expertise with using URIs when browsing or authoring. Web browsers are the ubiquitous way that people use URIs, and even in authoring tools, the primary mental model people have of URIs is derived from their use in browsers. In designing Swoop, we took the Web browser as our user interface (UI) paradigm, believing that URIs are central to the understanding and construction of Web ontologies. We contrast this to other ontology editors such as Protégé (Noy et
SemDiff: An Approach to Detecting Semantic Changes to Ontologies
www.igi-global.com/article/semdiff-approach-detecting-semantic-changes/2825?camid=4v1a

A Theory of Social Agentivity and its Integration into the Descriptive Ontology for Linguistic and Cognitive Engineering
www.igi-global.com/chapter/theory-social-agentivity-its-integration/76170?camid=4v1a