Appendix

Appendix

Appendix on Implementation: Languages, Editors, Reasoners, Browsers, Tools for Reuse

Web Ontology Language (OWL)

As we saw, the Web Ontology Language (OWL) is a family of knowledge representation languages that are the basic computer-implementable languages in which ontologies are written. OWL languages are based on description logics. Whereas first-order logic is designed to capture the formal structure of sentences and to capture in great detail the inferential relations that obtain among them, description logics are designed to provide more information (material as well as formal) about the contents of propositions and concepts while expressing fewer of the inferential relations among them. The ability to express more information makes description logics more useful for knowledge representation and ontology purposes, while the limited inferential information that they represent makes them more tractable for computers.

Editors: Protégé and OBO Editor

Because OWL programing languages and description logics are somewhat complicated to interact with directly ontology authoring and editing software has been developed. The most widely used tool of this sort is Protégé, which uses OWL and is based on the Java computer language. Rather than requiring users to be intimately familiar with the syntax and commands of some particular version of OWL, Protégé provides a uniform ontology viewing and editing experience that allows users to select commands, relations, and so on, from drop boxes and to enter definitions and other information in ordinary language. This software is freely available, along with tutorials and other user information, at http://protege.stanford.edu/. Another significant ontology editor is OBO-Edit, which is optimized for reading and writing ontologies in the OBO biological ontology file format. More information about OBO-Edit is available at <http://oboedit.org>.

Reasoners

Semantic reasoners (or simply *reasoners*) are pieces of software designed to infer the consequences of assertions in an ontology. Once an ontology has been implemented in some OWL language—through an editor such as Protégé or OBO-Edit—much of the information that has been entered in the ontology has an explicit structure that can be understood as an axiom or assertion. Reasoners use logical rules to draw conclusions from such axioms or assertions. This means that from the assertions in an ontology a reasoning program can determine new relations between universals or classes in the ontology (for instance, by tracing *is\_a* or *part\_of* relations through the ontology based on their transitivity). Such reasoners can be used to check for consistency of the information entered into the ontology. Reasoners in standard use with OWL include: Pellet (<http://clarkparsia.com/pellet/>), FaCT++ (<http://owl.man.ac.uk/factplusplus/>), and HermiT Owl Reasoner (<http://hermit-reasoner.com/>).

Browsers

Ontology browsers are tools that allow a user to visualize and explore single ontologies and also to query and compare a number of ontologies simultaneously. A very comprehensive ontology browser for biological and biomedical purposes is Bioportal (<http://bioportal.bioontology.org>), which allows users to search for relevant terms across and to browse in a very large and ever-expanding repository of existing ontologies. Another is the University of Michigan Medical School’s Ontobee (<http://www.ontobee.org/>), which is the default linked ontology data server for OBO Foundry library ontologies. As noted on their website, Ontobee is “aimed to facilitate ontology data sharing, visualization, query, integration, and analysis. Ontobee dynamically dereferences and presents individual ontology term URIs to (i) *HTML web pages* for user-friendly web browsing and navigation, and to (ii) *RDF source code* for Semantic Web applications.”1 Ontology browsers are important for purposes of ontology design because they allow would-be designers to survey existing ontologies and so as to avoid reinventing the wheel. By making it possible to compare the entry for a single term across multiple ontologies, they also make it possible to do quality control, checking to see which ontology has the best or most scientifically accurate information. Other browsers include Ontology-Browser (<https://code.google.com/p/ontology-browser/>), jOWL (<http://jowl.ontologyonline.org/>), and QuickGo, a browser for Gene Ontology terms and annotations (<http://www.ebi.ac.uk/QuickGO/>).

Tools for Reuse: Ontofox and Mireot

While ontology browsers make the task of searching for and reusing existing ontologies easier, by themselves they still leave a potential user with the task of finding some way to copy the relevant data from existing ontologies into their own ontologies. In response to this problem, researchers have begun developing software the specific goal of which is to copy complete parts of existing ontologies (perhaps discovered via an ontology browser) for use elsewhere. For example, OntoFox is a project driven by the principles of MIREOT (minimum information to reference an external ontology term), which have as their aim making it both possible and efficient for application ontology designers to utilize preexisting reference ontologies in their application designs rather than creating new content for each new application. See Melanie Courtot et al., “MIREOT: The Minimum Information to Reference an External Ontology Term,” *Applied Ontology* 6, no. 1 (January 2011): 23–33. Also, the OntoFox tool (<http://ontofox.hegroup.org/introduction.php>) can retrieve selected portions of taxonomies from already-existing ontologies based on a specification of the highest (most general) and lowest (most specific) terms from some section of the ontology that are of interest to the searcher.

{Notes\_begin}

Appendix

1. Accessed September 1, 2014, <http://www.ontobee.org/>.

{Notes\_end}