Towards ubiquitous OWL computing: Simplifying programmatic authoring of and querying with OWL axioms

Hilmar Lapp and Jim Balhoff, National Evolutionary Synthesis Center (NESCent), Durham, NC
Email: hlapp@nescent.org

Semantic web technologies have enjoyed a growing popularity in integrating and connecting data, especially in the life sciences. For example, RDF and triple stores have been used to connect large amounts of diverse biological data and knowledge by shared entities and properties. Ontologies have been applied with great success to harmonizing and defining terminologies for many areas of descriptive biology, most prominently gene function, and the processes and locations in which gene products act. OWL reasoning and OWL ontologies have been used to assess the similarity of biological observations described in natural language text by the degree they share semantics. Even though triple store, ontology authoring, and reasoning technologies have become substantially more powerful in recent years, applying them at scale and for complex discovery applications is still hampered by informatics challenges and limitations. Here, we present two generically useful tools that were developed in response to ontology computing challenges encountered within the Phenoscape project (http://phenoscape.org), an initiative that uses semantic web technologies to render evolutionary phenotype descriptions amenable to computational data mining and integration.

The first tool, named Scowl, greatly simplifies and thereby accelerates the kind of ad-hoc, in bulk OWL ontology axiom authoring that is often necessary as part of the build pipelines for OWL ontology-based knowledgebases. Specifically, Scowl provides a declarative API for creating OWL class expressions and axioms in a natural way that mirrors OWL Manchester Syntax. The tool was originally designed to allow better authoring and expert review of the axiom generations necessary for transforming datasets into pertinent OWL models. Due to the high human readability of OWL Manchester Syntax it could also be used for literate programming of ontologies in ways that much better support revision control, integration testing, and other collaborative authoring infrastructure available for source code-like text.

Owlet, the second tool, addresses the issue that the expressivity of reasoners built into RDF triple stores is much more limited compared to OWL reasoners. As a consequence, using OWL constructs more complex than simple named classes, such as disjunctive OWL class expressions composed ad-hoc in response to user input, often result in complex, error prone, and poorly performing queries in SPARQL (the query language for RDF), whereas an OWL reasoner could resolve them quickly and correctly. Owlet allows integrating OWL class expressions directly into SPARQL queries. It masquerades as a SPARQL query endpoint, recognizes an OWL expression embedded in OWL Manchester Syntax, expands the expression to a FILTER clause that enumerates the matching ontology classes, and passes the resulting expanded query on to the SPARQL endpoint for the triple store.

Both Scowl and Owlet are written in Scala, and available under the MIT License from Github at http://github.com/phenoscape/scowl and http://github.com/phenoscape/owlet, respectively.