Using Semantic Web Modeling and Technology for Automated Search for Data

Nancy Wiegand, University of Wisconsin-Madison, wiegand@cs.wisc.edu

Searching for data sources over the Web, including using portals that record metadata, is problematic. User entered keywords often do not return needed data sources or return many irrelevant results. Instead, data specialists may resort to general Internet searches or a sequence of phone calls. Better methods are needed to retrieve data, and, ideally, the process would be automated. We explore the abilities and limitations of Semantic Web languages, tools, and rule engines for this purpose.

We observe that many searches are for tasks that need the same types of data with just the location varying. For example, tasks involving emergency response or land use planning need similar data but for different places. To solve the problem of finding data sources and to automate the process, we create a model to formalize a knowledge base and associations within it. Our model is task-based and includes developing an ontology of tasks, derived from the symbology of incidents developed by the Homeland Security Working Group, and an ontology of data sources derived from FGDC framework data. We also import the FGDC OWL metadata ontology and include a place taxonomy. Correspondences between types of data sources needed for types of tasks are established using examples from emerging reports of emergency task forces. This information is then formally recorded using the OWL Web ontology language within Protégé, an ontology editor. Instance data is added for evaluation.

Although we attempted to fully design our model using OWL restrictions to keep all information within an established knowledge base, we found that geospatial data have too many selection criteria, such as scale, date, and accuracy, for that to be feasible. That is, full modeling by declaring numerous restrictions using OWL's "someValuesFrom" construct within anonymous subclasses was very cumbersome and inflexible. However, the base relationships between types of tasks and data sources were still modeled in this manner to avoid hard-coding criteria in rules or queries as much as possible. Also, for simplicity, we expressed base relationships using a "needs" object property rather than creating an overarching class as done in various other Protégé applications. An example someValuesFrom restriction for a fire emergency is "∀ needs (Road U LandCover U Hydrography)".

To specify further selection criteria, we needed to augment our system with Jess, a rule engine, which is accessible in Protégé through the JessTab plugin. Jess rules allow any number of selections and the use of comparison operators. However, due to the lack of JessTab's ability to translate OWL object properties to Jess assertions, we had to create general instance data for class types and re-set restriction information using instances. This then did allow the inferencing we needed.

In summary, our user interface allows selection of a task from the task ontology and a location from a place ontology. Through inferencing over base relationships modeled as OWL restrictions, the types of data sources needed for the selected task are returned. Using further rules that are pre-established, additional criteria are specified for each data source type, allowing the return of URLs to needed data sources.

The advantage of using ontologies rather than a database query approach is that base information can be re-used for other applications. The advantage of a formalized and automated system is that, in an emergency for example, anyone can quickly retrieve needed data sources. Further, a full expert system could execute subsequent rules if particular criteria from initial rules do not yield a data source (e.g., to find a later year or smaller scale). Such a system can be employed within existing portals or stand-alone on the Web using emerging distributed technologies including OWL-QL.