

Knowledge is just a click away: Semantic Query Answering in DSpace*

Dimitrios A. Koutsomitropoulos¹, Ricardo Borillo Domenech², Georgia D. Solomou¹, Aikaterini K. Kalou¹ and Theodore S. Papatheodorou¹

¹ High Performance Information Systems Laboratory (HPCLab),
Computer Engineering and Informatics Dpt., School of Engineering,
University of Patras, Buidling B, 26500 Patras-Rio, Greece

² Servicio de Informática, Universitat Jaume I,
Rectorado, 12071, Castellón, Spain
kotsomit@hpclab.ceid.upatras.gr, borillo@uji.es
solomou@hpclab.ceid.upatras.gr

Abstract. Information and knowledge retrieval are today some of the main assets of the Semantic Web. However, a notable immaturity still exists, as to what tools, methods and standards may be used to effectively achieve these goals. No matter what approach is actually followed, querying Semantic Web information often requires deep knowledge of the ontological syntax, the querying protocol and the knowledge base structure as well as a careful elaboration of the query itself, in order to extract the desired results. In this paper, we propose a structured semantic query interface that helps to construct and submit entailment-based queries in an intuitive way. It is designed so as to capture the meaning of the intended user query, regardless of the formalism actually being used, and to transparently formulate one in reasoner-compatible format. This interface has been deployed on top of the semantic search prototype of the DSpace digital repository system.

Keywords: Semantic Web, queries, ontologies, entailment, guided input.

1 Introduction

This demonstration will present a *structured* querying mechanism and interface that helps to construct and submit entailment-based queries to web ontology documents. The main idea is to aid the user in breaking down his intended query expression into several *atoms*. These atoms are then combined to form allowed expressions in Manchester Syntax, as the closest to our purposes regarding user-friendliness. At the same time, the interface tries to be as intuitive as possible by automatically disallowing (graying out) nonsensical combinations (for example, select

* This demonstration is based on the paper “A Structured Semantic Query Interface for Reasoning-Based Search and Retrieval”, to be presented at ESWC 2011.

a restriction without selecting a property first), offering dynamic auto-complete choices and classify them as per class (type) or relation, disclosing namespace prefixes when possible, marking the various fields with NL-like labels and presenting results based on their class or type. To the best of our knowledge no other system using DL-based query languages exists, that follows the idea of controlled input forms for the structured formulation of semantic queries.

Based on this idea we have developed a prototype application as an add-on to the DSpace digital repository system, latest version (1.6.2 and 1.7.0)¹. This work builds upon and evolves earlier efforts for creating a semantic search service for DSpace [2]. The novel semantic search interface is backed up by a new *DSpace Semantic API* that supports a pluggable design for reasoners as well as OWL 2.0 and the newest OWL API v.3. Most importantly, our Semantic API is designed along the same principles as its predecessor, i.e. to remain independent of the DSpace business logic and to be agnostic to the rest of the user interface or even the underlying ontology.

Our prototype is openly available at: <http://apollo.hpclab.ceid.upatras.gr:8000/jspui16/semantic-search>. Source code is maintained at: <http://code.google.com/p/dspace-semantic-search/> where instructions and latest developments can be found.

2 Functionality

When the semantic search interface is loaded, one can distinguish among three separate tabs: *Search* (default), *Advanced topics* and *Options*.

The figure shows two parts of the semantic search interface. The top part is the 'Search' tab, which includes a 'Search for' field with a 'not' checkbox, a 'Restriction' dropdown, an 'Expression' field, a 'Condition' section with 'and' and 'or' radio buttons, an 'Add term' button, a 'Generated query' field, and 'Search' and 'Clear query' buttons. The bottom part is the 'Options' tab, which includes a 'Reasoner' dropdown (set to 'Pellet') and an 'Ontology' text field (containing a URL), with a 'Save options' button. To the right, a detailed view of the 'Restriction' dropdown menu is shown, listing options: 'some', 'min', 'max', 'only', 'value', and 'exactly'.

Fig. 1. Search and Option tab of the semantic search interface.

The *Search* form contains all necessary elements for guiding users in building queries in Manchester syntax as intuitively as possible. Each component in this form

¹ <http://www.dspace.org/>

corresponds to a certain building atom of the query expression. Their functionality is described in detail later in this section.

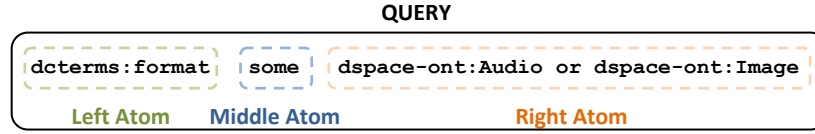


Fig. 2. The building atoms of a query expression in Manchester syntax.

The *Advanced topics* tab is currently inactive and is reserved for future extensions of the system, like for example the support of other query syntaxes, such as SPARQL.

The *Options* tab includes options that allow the user to change the ontology against which they perform their search, as well as the underlying reasoner (currently, Pellet or FaCT++). For altering the knowledge base, we only need to supply a valid URL of an OWL/OWL 2-compliant ontology. In addition, the user can switch between reasoners dynamically (Section 3.4).

Next we describe the various elements of the *Search* tab, according to their number in Fig. 1. Based on Manchester syntax's primitives for formulating an expression [1] and depending on the values entered by the user in each preceding field, subsequent fields are enabled or disabled accordingly. Figure 2 depicts the three main atoms of such a query expression. What is more, an auto-complete mechanism is enabled where necessary, for guiding users in supplementing information.

1. **Search for:** It corresponds to the outmost left (first) atom of a Manchester syntax expression. This can be either a property name or a class name. An auto-complete mechanism is triggered as soon as a word starts being typed, suggesting names for classes and properties that exist in the loaded ontology. For users' convenience, suggested values have been grouped under the title *Types* (for classes) and *Relations* (for properties) (left part of Fig. 3). The check box is used for declaring the negation of the class expression that starts being constructed. For simplicity, all prefixes are kept hidden from users and the system is responsible for adding them automatically, during the query generation process. The following two fields are not activated, unless a property name has been selected in this step.
2. **Restriction:** This represents the middle atom of the expression. Provided that a property is entered in the previous field, a number-, value- or existential restriction should now be set. Hence, the "Restriction" drop down menu becomes active, containing respective Manchester syntax keywords.
3. **Expression:** This is a free-text field where the user can supply a single class name or expression (quantification), an individual (value restriction) or a number, optionally followed by a class (cardinality restriction). An auto-complete facility is provided for class names. This forms the outmost right (last) atom of the query expression.
4. **Condition:** From now on, the user can recursively construct more class expressions, combining them in conjunctions (*and*) or disjunctions (*or*). Consequently an appropriate *Condition* should be set for expressing the type of logical connection.

5. **Generated Query:** This field gradually collects the various user selections and inputs, ultimately containing the final query expression. It is worth noting that this is an editable text box, meaning that expert users can always bypass the construction mechanism and use it directly for typing their query.
6. **Add term:** Adds a completed expression to the *Generated Query* field. This also checks if the expression to be added is valid, and pops an error message otherwise.
7. **Search:** When pressed, evaluates the query expression as it appears in the *Generated Query* field. It also clears all other fields, thus giving the user the opportunity to further refine his initial query.
8. **Clear query:** Clears the form and makes it ready to accept new values.

Once the query has been evaluated, obtained results appear right below the search form. They are organized in the form of a two-column table, as shown in Fig. 3: *Value* contains the retrieved entities, whereas *Type* indicates at least one of the classes to which each entity belongs, thus providing users with a hint about their type. All retrieved entities are clickable and when selected, a separate page is loaded, containing the complete ontological information about the clicked entity. More details about this page and its elements can be found in [2].

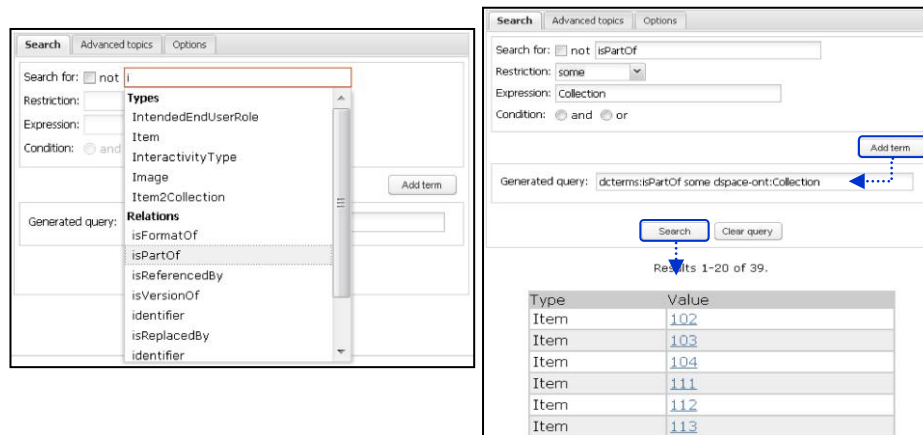


Fig. 3. The auto-complete mechanism and the results table in the semantic search interface.

3 Demonstration

The purpose of this demonstration is to let end-users and conference participants to have a direct hands-on experience with the proposed interface. At the same time, it is expected to receive feedback that may help to evaluate and improve the system.

The demonstration methodology will be the one of a *controlled experiment*: Demo participants will be engaged in a series of predefined query scenarios on the DSpace ontology or another ontology document suitable for this purpose. After an introduction with examples and some initial guidance, participants will be asked to construct queries on their own, thus exploring and understanding the interface's

features through practice. In addition, users will be prompted to repeat these queries, using this time the traditional keyword-based mechanism, and compare the results. At some point, users will realize that some queries are incomplete or impossible to be expressed using keywords only, therefore the practical gains of reasoning-based querying will also be shown.

At the end of the session, participants will be asked to anonymously fill in a questionnaire with a series of mostly closed-format questions, like for example: "How familiar are you with Manchester syntax?", "Did you find what you were looking for?", "How easy was for you to formulate the query?", "How does it compare with the key-based search facility?", "How many queries did you actually fulfill?" and so on. Collected feedback will be useful for evaluating the participants experience and the system itself.

The technical configuration will include deploying DSpace with the semantic search interface on a server and set it up with the experimental ontology. Provided that there is network availability on premises, people will be asked to connect to our server from their own laptops. Internet connection is preferred but not required. Guided scenarios will be presented and explained through a projector. The demo is expected to last from three to five quarters of an hour.

References

1. Horridge, M., Patel-Schneider, P.S.: OWL 2 Web Ontology Language: Manchester Syntax. W3C Working Group Note. (2009) <http://www.w3.org/TR/owl2-manchester-syntax/>
2. Koutsomitropoulos, D., Solomou, G., Alexopoulos, A., Papatheodorou, Th.: Digital Repositories and the Semantic Web: Semantic Search and Navigation for DSpace. In: 4th International Conference on Open Repositories. Atlanta (2009)