DSpace Semantic Search v2.0: What's New and Current Status

Dimitrios A. Koutsomitropoulos[1], Georgia D. Solomou[1] and Ricardo Borillo Domenech[2]

[1]High Performance Information Systems Laboratory, School of Engineering, Computer Engineering and Informatics Department, University of Patras,
Building B, 26500, Patras-Rio, Greece
[2]Servicio de Informática, Universitat Jaume I,
Rectorado, 12071, Castellón, Spain
kotsomit@hpclab.ceid.upatras.gr, solomou@hpclab.ceid.upatras.gr
borillo@uji.es

Abstract. DSpace Semantic Search v2.0 is the next version of the reasoning-based querying and navigation service for the DSpace digital repository system. Compared to its predecessor, v2.0 has been significantly refactored and comes with an improved interface, additional functionality and a modular, 'add-on' architecture. In this paper, we introduce this service to the community by giving an abridged account of new and upcoming features and give pointers for further information and involvement.

1 Introduction - The story so far

The semantic search and navigation service is the outcome of efforts to enable semantic searching capabilities for any web-based digital repository system. A first prototype [4] was developed as an add-on for DSpace version 1.4.2 and since then, several enhancements have been incorporated, upgrading both its user-friendliness and functionality.

This initial version of the semantic search interface ('*Semantic Search* v1.0*'*) simply amounted to a text field where users had to type in their queries, following the principles of the Manchester OWL Syntax [2]. Search was conducted against the *DSpace ontology* (populated automatically through OAI) and results were displayed in the form of a browsable list. Users were responsible for typing their queries correctly, so this first prototype was considered a bit awkward, especially for those not familiar with Manchester Syntax.

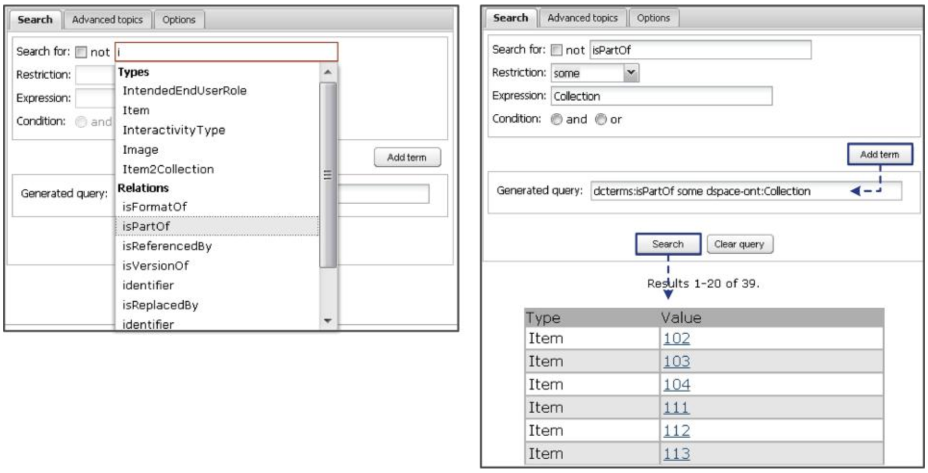
With Semantic Search v2.0, a structured querying mechanism and interface is introduced, that guides users in building semantic queries. At the same time, the interface tries to be as intuitive as possible by making suggestions, checking user input and enabling or disabling subsequent fields accordingly.

The novel semantic search interface is backed by a new *DSpace Semantic API* that sports a pluggable design for reasoners as well as proper handling of OWL 2 and is based on the newest OWL API v3 [1]. Most importantly, this 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 to the underlying ontology.

Semantic Search v2.0 is hosted and maintained as a Google Code project (http://code.google.com/p/dspace-semantic-search/). From here, one can checkout, deploy and try this service, contribute to open issues and raise his opinion. What is more, there is an ongoing online survey, available at http://goo.gl/Zt0aP, where anyone can take part and evaluate several aspects of this service. A more thorough account of the architecture, methodology and design principles behind Semantic Search v2.0 can be found in [3].

2 New Features

Since its first launch, the semantic search service has been improved and augmented with new elements, aiming to enhance its performance and make searching an easier and more intuitive experience. Here is a summary of new features:



Search for: □ not i
Restriction:
Expression:
Condition: ○ and
Generated query:

Search for: □ not isPartOf
Restriction: some
Expression: Collection
Condition: ○ and ○ or
Generated query: dcterms:isPartOf some dspace-ont:Collection

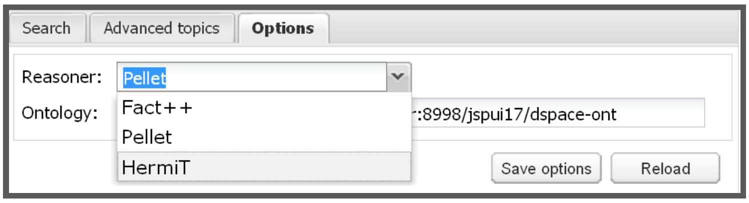
Results 1-20 of 39.

Type	Value
Item	102
Item	103
Item	104
Item	111
Item	112
Item	113

Fig. 1. The auto-complete and query construction mechanisms of the semantic search interface.

New Interface: The old single text box has been transformed into a structured query interface, consisting of a group of input fields and drop-down menus. These fields correspond to the building blocks of a Manchester Syntax expression (*query atoms*). In addition, the new interface guides users in the construction of legit Manchester Syntax queries by automatically disallowing (graying out) nonsensical combinations (for example, select 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 (see Fig. 1). More tabs have also been added: *Options* allows selection of reasoner and ontology, while *Advanced topics* is reserved for future extensions (e.g. SPARQL).

Hot-Swap between reasoners: One of the design principles of Semantic Search was extensibility and support for different reasoners and ontologies. Therefore, the new Semantic API features the ability to "hot-swap" between reasoners dynamically (see Fig. 2). For the time being, any OWL API compliant reasoner can be supported, including out-of-the-box support for Pellet, FaCT++ and HermiT.



Reasoner: Pellet

Ontology: :8998/jspui17/dspace-ont

Fig. 2. Changing reasoner through the *Options* tab.

Ontology caching: Semantic Search runs by default on top of the DSpace ontology. Nevertheless it can work against any OWL ontology accessible through the Web. Loaded ontologies can be reused, thus avoiding the overhead of reloading and parsing the whole ontology definition. In particular, when a user asks for a new ontology, it is loaded and stored only once in an internal registry. When another user asks for the same ontology, no re-parsing is needed, and the ontology is served from the registry.

Precomputing inferences: An important feature implemented in current releases of Semantic Search v2.0 is inferences' precomputation. Precomputation is an initialization phase where the reasoner classifies the ontology, and computes in advance certain reasoning operations. Precomputation results are then readily available in-memory and, instead of engaging the inference engine each and every time, new queries can take advantage of them, resulting in a significant performance increase.

Ontology enhancements: Semantic Search v.2.0 includes several enhancements and/or fixes throughout the DSpace ontology creation process as well as in the involved ontology documents, aiming for better resource discovery, reasoner compatibility and extended inference capabilities. For example, the Unique Name Assumption (UNA) is now enforced, thus allowing queries that constrain the number of entities related through a certain property, e.g. the number of items produced by the same author (number restrictions).

3 Current Status and Upcoming Features

Semantic Search v2.0 is already available as a DSpace add-on (http://www.dspace.org/addons/#semantic). To apply the service, code patches are available at Google Code, currently compatible and tested for DSpace versions 1.6.2 onwards. Alternatively, users can check out the modified DSpace version directly from Google Code's SVN trunk folder. The Google Code page contains also a concise fact-sheet about the semantic search service, deployment instructions that help installing the service and configuring reasoners, as well as detailed documentation about how to use the service.

DSpace Semantic Search is under ongoing development. Upcoming version 2.1 has just been released and is compatible with DSpace 1.8. Some additional features included in this release are:

- *Support for more OWL API compliant reasoners* (like HermiT).
- *Adoption of the latest OWL API v3.2.4*, coming with additional capabilities for reasoner and ontology handling.
- *Reasoner caching*, meaning that, just like ontologies, reasoner objects among different users can also be cached in an internal registry and reused. Combined with inference precomputation, this can also improve average performance of multiple individual requests on the same ontology.
- *A 'Reload' button* that would clear registry and reload both the ontology and the reasoner.

Enhancements for future versions currently under consideration include:

- Interface fine-tuning based also on the user survey results (e.g., support for advanced Manchester Syntax constructs like facets and inverse properties).
- Add the ability to remember and predict ontology URLs, based on recent history.
- More informative results rendering, e.g. by including the full list of membership classes as a tooltip.
- Scalable architecture based on triple-stores.
- Add support for expressing queries in other syntaxes as well, e.g. SPARQL-DL [4].

Finally a dedicated mailing list for the project has just been set up and is publicly available as a Google Group (http://groups.google.com/group/dspace-semantic-search). Latest patches and new features will be announced in this list, and users can post their suggestions, questions and feedback.

4 Evaluation process

To evaluate Semantic Search, we have prepared and are now conducting an online survey that explores both the service's usability and performance. Our goal is first to discover the service's impact and appeal to all kinds of users (those familiar and non-familiar with Semantic Web concepts and Manchester Syntax), and then to detect possible areas for improvement and collect user suggestions. In addition to the survey, an analysis of several real queries based on logged data is also considered, in accordance to current recommendations for the evaluation of semantic search systems (e.g., [7], [6]).

First-circle survey results reveal that users tend to prefer the guided query mechanism with its auto-complete capabilities. Actually, they find that this service makes semantic searching easier and that it is better than traditional keyword-based search in terms of accuracy or retrieval capabilities. On the other hand, query analysis indicates a low rate for failed queries as well as a significant percentage of complex, restriction-based expressions that resulted into fine-grained retrieval.

The online survey is still up and running and DSpace community members as well as any other interested party are more than welcome to participate, make suggestions and thus contribute to Semantic Search evaluation and further development. Users can have a hands-on experience with the system at our demo installation, available at: (http://apollo.hpclab.ceid.upatras.gr:8000/jspui17-demo/). This demo is based on DSpace 1.7.1 and includes some 50 items containing abstracts and other material from the ESWC series of conferences (http://www.eswc2011.org/).

References

1. Horridge, M., Bechhofer, S.: The OWL API: A Java API for Working with OWL 2 Ontologies. In: 6th OWL Experiences and Directions Workshop. Chantilly, Virginia (2009)
2. Horridge, M., Patel-Schneider, P.S.: Manchester Syntax for OWL 1.1. In: 4th OWL Experiences and Directions Workshop. Gaithersburg, Maryland (2008)
3. Koutsomitropoulos, D., Borillo Domenech, R., Solomou, G.: A Structured Semantic Query Interface for Reasoning-Based Search and Retrieval. In: 8th Extended Semantic Web Conference. Heraklion, Greece (2011)
4. Koutsomitropoulos, D., Solomou, G., Alexopoulos, A., Papatheodorou, T.: Digital Repositories and the Semantic Web: Semantic Search and Navigation for DSpace. In: 4th International Conference on Open Repositories. Atlanta, Georgia (2009)
5. Sirin, E., Parsia, B.: SPARQL-DL: SPARQL Query for OWL-DL. In: 3rd OWL Experiences and Directions Workshop. Austria (2007)
6. Uren, V., Sabou, M., Motta, E., Fernandez, M., Lopez, V., Lei, Y.: Reflections on five years of evaluating semantic search systems. In: International Journal of Metadata, Semantics and Ontologies, 5(2), 87-98 (2011)
7. Wrigley, S., Reinhard, D., Elbedweihy, K., Bernstein, A., Ciravegna, F.: Methodology and campaign design for the evaluation of semantic search tools. In: Semantic Search 2010 Workshop at World Wide Web Conference. Raleigh, NC (2010)