KNOWLEDGE MANAGEMENT AND ACQUISITION IN DIGITAL REPOSITORIES

A Semantic Web Perspective

Dimitrios A. Koutsomitropoulos, Georgia D. Solomou, Andreas D. Alexopoulos and Theodore S. Papatheodorou

High Performance Information Systems Laboratory, School of Engineering, Dpt. of Computer Engineering and Informatics, University of Patras, Building B, 26500, Patras-Rio, Greece

Keywords: Knowledge Management, Metadata, Interoperability, Ontologies, Reasoning, Digital Repositories.

Abstract: Information management, description and discovery, as they are today implemented in digital repositories and digital libraries systems, can surely benefit from the stack of Semantic Web technologies. Most importantly, the ability to infer implied information over declared facts and assertions, based on their rich descriptions and associations, can span new possibilities in how stored assets can be accessed, searched and discovered. In this paper we propose a process and implementation that provides for inference-based knowledge discovery, retrieval and navigation on top of digital repositories, based on existing metadata and other semi-structured information. We show that it is possible to produce added-value and meaningful results even when existing descriptions are only flatly organized and we achieve this with little manual intervention. Our work and results are based on real-world data and applied on the official University of Patras institutional repository that is based on DSpace.

1 INTRODUCTION

The Semantic Web infrastructure mainly relies on specifications for expressing ontologies in web-compatible format, like OWL or OWL 2 (Grau, et al., 2008) and, lately, on programming efforts for manipulating such ontologies, like the OWL API (Horridge, et al., 2007) and the Protégé 4.0 code-base (the CO-ODE project, http://www.co-ode.org/). To be able to fully reap the benefits of a *Semantic* Web, reasoning over ontological information is of exceeding importance, a fact that was sometimes overlooked in the past, possibly because of the immaturity of available tools and techniques. To our belief, the ability to infer implied information over declared facts and assertions is one of the most prominent reasons to investigate and implement the Semantic Web, in a sense that adds an "AI" flavor to the current Web (Hendler, 2008).

Therefore, in this paper we present and document a process that builds upon the well-known *digital repositories* paradigm and enhances it with the Semantic Web's features. The main goal that drives our efforts is not to re-implement a digital repository system using Semantic Web APIs and technologies, but to provide inference-based knowledge discovery, retrieval and navigation *on top* of such a system, based on existing metadata and other semi-structured information.

To prove our concept, we describe a concrete, working prototype that provides for inference-based search and navigation on top of the DSpace digital repository system. DSpace has become a popular open-source digital repository solution with one of the most rapidly growing user bases worldwide. DSpace metadata follow the Dublin Core (DC) specification by default, while it is possible to import and use other metadata schemata as well.

This paper is further organized as follows: First an overview of related work on digital libraries and semantics is given; then, we introduce the process for constructing the repository's ontology and point out its most important aspects. Following, we document our extensions to the DSpace system and the implementation of the ontology management, search, navigation and reasoning services. Finally we give specific examples and results that demonstrate these new capabilities and summarize the conclusions of our work. A partial description of

Koutsomitropoulos D., Solomou G., Alexopoulos A. and Papatheodorou T. (2009).
KNOWLEDGE MANAGEMENT AND ACQUISITION IN DIGITAL REPOSITORIES - A Semantic Web Perspective.
In *Proceedings of the International Conference on Knowledge Management and Information Sharing*, pages 117-122

this work and source code are freely available at: http://wiki.dspace.org/ index.php/User:Kotsomit.

2 RELATED WORK

Some widely adopted mechanisms for digital repositories that, similar to our work, appear to utilize Semantic Web technologies are BRICKS, SIMILE, Fedora and JeromeDL plus the more recently appeared Talia.

Both BRICKS (Risse, et al., 2005) and Fedora (http://www.fedora.info) provide the basic architecture upon which digital library applications can be developed and deployed. Their functionality is further enhanced by supporting some basic Semantic Web technologies, like the expression of relations between objects in RDF and the retrieval of data through the evaluation of queries using SPARQL or other RDF query languages.

Through the application of RDF and Semantic Web techniques, SIMILE (http://simile.mit.edu/) offers DSpace improved support for arbitrary schemata and metadata and provides an architecture for disseminating digital assets. Among SIMILE's implemented tools, the one that pertains mostly to our work is a faceted browser, known as Longwell (its DSpace version is called Dwell), that gives the ability to cross-section the data along fixed dimensions of structured metadata. Furthermore, SIMILE provides tools to merge lexical or semantic variants via simple inferencing. Nevertheless, the current implementation of SIMILE does not seem to offer reasoning based querying.

JeromeDL (Kruk, et al., 2005) is a "social semantic digital library" that stores its metadata in RDF all along, by utilizing a corresponding RDF store (Sesame). A level of inference is supported through a simple recommendation engine based on Prolog. JeromeDL seems to solve the "semantic bootstrapping" problem following a bottom-up approach, since the ontological schema is constructed and populated in advance. In such a way the problem of retrieving semantic implications and inference-based results, also from flatly organized relational data base sources, is circumvented.

Finally Talia (Nucci, et al., 2008) is a library platform that stores its metadata into a relational DB schema and keeps it "in sync" with an RDF data store (Redland). In addition, it provides a unified query interface for both database and RDF metadata using SQL or SPARQL, as required. However, Talia does not do any inferencing on the RDF data and leaves this responsibility to the underlying RDF store.

3 CREATION AND POPULATION OF THE ONTOLOGICAL MODEL

In this section we give a brief outline of how we have developed a Semantic Web ontology out of the repository's metadata, based on which we can employ our complementary, semantics-aware services. A more detailed description of this process is out of the scope of this paper and can be sought in (Koutsomitropoulos, et. al., 2008b, 2009b).

Based on the DC RDF(S) schema (Nilsson, et al., 2008) we have developed a *semantic application profile* (Koutsomitropoulos, et al., 2009a) in three main steps:

- First, we transferred the DC original schema in OWL format.
- Then we augmented its semantics, by using property characteristics not available in RDFS: for example, we have identified some DC properties to be inverse, symmetric or transitive and declared them as such.
- We further profiled the model by including refinements for our particular application, that is, the University of Patras digital repository. We have modelled vocabularies in taxonomies, introduced new properties for DSpace relations ('author', 'sponsorship') and new classes for DSpace notions ('item', 'collection') that the original DC does not provide for. Further, we used OWL 2-specific constructs, like role-chains, to represent intrinsic complex relations, like the 'co-author' relationship between authors.

A naive attempt to model the DC domain as thorough as possible, by representing each and every potential semantic relationship, can easily render the ontology undecidable (Koutsomitropoulos, et al., 2008b). However we have identified that the *punning* feature, introduced with OWL 2, can reasonably deal with ambiguities and meta-modelling requirements, inherent in the DC specification.

The resulting ontology, including the new refinements, is then populated in an automated way from metadata already existing within the live DSpace installation of the University of Patras institutional repository. These metadata are harvested through the repository's OAI-PMH interface (Lagoze, et al., 2002) and mapped to the ontology using an XSLT developed for this purpose.

4 SEMANTIC SEARCH AND NAVIGATION

In this section we discuss the design decisions and the implementation of the semantic enhancements to the DSpace digital repository system.

4.1 Design Goals

Most of the design decisions stem from a set of requirements that where posed beforehand, in order to guarantee reproducibility and applicability of our efforts, as well as to ensure the potential of the semantic services offered. They can be summarized as follows:

Interoperability. At its core, interoperability is achieved by adhering to information standards, at any level: Repository's metadata are structured according to the XML format, ontology models are represented using W3C's OWL and OWL 2 specifications and the semantic gap between the two is bridged using an XSLT transformation. Further, this transformation upgrades interoperability to a semantic level, by rendering the repository's metadata descriptions semantically compatible to the ontology's structures. Lacking a communication protocol for the exchange of OWL information, we at least opt for wrapping and transforming OAI-PMH responses that offer a standard way for harvesting metadata from data providers. This comes in contrast to accessing the repository's database directly, as this could be dependent on the proprietary DB schema. As a result, our approach could be seamlessly integrated with other resource management systems, at least OAI compliant ones.

Support for OWL 2. The need to support this newly proposed extension to OWL comes from the fact that OWL 2 is able to represent a richer set of semantics than its predecessor, thus enabling more advanced inferences. On the other hand, this reduces our choices of inference engines to only supporting ones, such as FaCT++ and Pellet. Performance is not our main concern here, since the OWL 2 reasoning algorithm is known to be scalable (Horrocks, et al., 2006) and its implementation in these reasoners is heavily optimized. However, we are forced to use a direct in-memory implementation, since none of these reasoners supports a communication interface other than DIG, which is currently incompatible with OWL DL, not mentioning OWL 2 (Koutsomitropoulos, et al., 2008a). Therefore, high expressivity comes at the cost of a truly distributed 3-tier architecture.

Extensibility. A major design decision was to totally implement our extensions using the OWL API, while avoiding references to DSpace specific methods and classes. OWL API equips us with a satisfactory layer of abstraction on top of which further extensions can be implemented. In addition, it does not restrict us to any particular inference engine or a specific reasoning approach: The selection of the reasoner class constructed can be easily parameterized, while the reasoning strategy can stay the same, rendering our implementation reasoner-independent. Furthermore, it is easy to support other querying protocols and/or methods: In our implementation, a query is given in the form of a Manchester Syntax class expression (Horridge & Patel-Schneider, 2008). Just as easily, query formulation can be extended to follow another paradigm, such as SPARQL/OWL (Sirin & Parsia, 2007), as soon as its specification grows mature and a supporting parser is implemented.

Figure 1: Semantic Search interface and the auto-complete facility.

User-friendliness/Intuitiveness. Unfortunately, a common way for querying OWL knowledge bases has not been standardized yet. SPARQL is a language for querying RDF graphs, but it does not take into account OWL's richer semantics. To build a "user-friendly" service and keep the complexity of query formulation as low as possible, we implemented a query interface based on class construction using Manchester Syntax, in a way inspired by the DL Query Tab of Protégé. Query results are formed by all individuals that are inferred to be instances of the constructed class. Manchester Syntax has the advantage to offer a pseudo-natural English language expression of classes, thus facilitating, to some extent, the end user to formulate a query. In addition, we have tried to make this process intuitive, by implementing an AJAX-based suggestion and auto-complete mechanism, where matching entities names are suggested to the user, as the query is typed in (Figure 1).

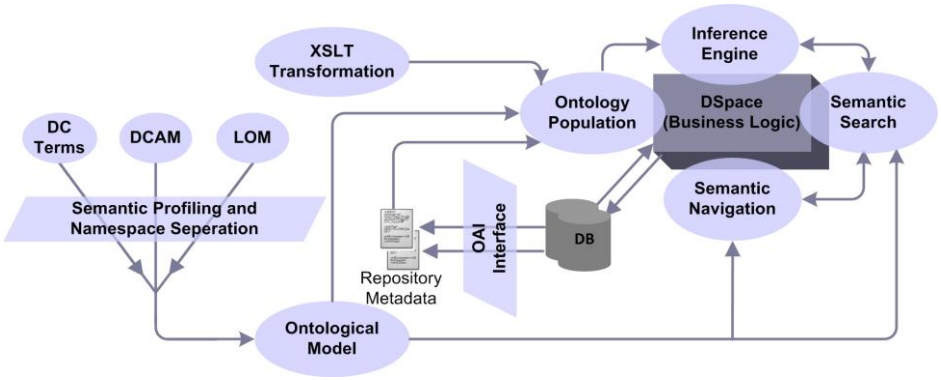

Figure 2: Architectural model of the Semantic Search extension of DSpace.

4.2 Architectural Overview

An overview of the services we have built around DSpace is depicted in Figure 2. The most important modules and interfaces that enable semantic services in our digital repository are the following:

- *Semantic Search* interface, which, in collaboration with the appropriate inference engine, allows for the construction, submission and evaluation of a semantic query. Retrieved results are displayed here in the form of a list.
- *Semantic Navigation* interface is where detailed ontological information about a selected entity (individual) is presented.
- *Ontology Population* refers to the dynamic construction of the ontology, which comes from DSpace's OAI harvested metadata, after applying the appropriate XSLT transformation on them.
- The *Inference Engine* is responsible for processing the ontological documents and for performing reasoning over them. We have chosen FaCT++ but any other DL reasoner may be used, as stated in section 4.1.

Context with DSpace itself is indirectly maintained, since it is still possible to open DSpace's simple item view page from within the navigation pane (Figure 3).

5 EVALUATION AND EXAMPLES

In this section we give some examples of how semantic-enabled search and navigation can work towards discovering and acquiring new and implied knowledge. This knowledge is impossible to be retrieved through a traditional querying interface, as there is not even a way to express such requests using solely combinations of matching keywords, let alone reasoning and inference themselves (Horrocks, 2008). Further, we see that these services allow retrieval and presentation of entities of any type, not just items.

5.1 Entity Retrieval

In DSpace, the main information unit is the *item*, which represents a specific resource (document, image or other) that has been uploaded in DSpace, as well as its containers, namely *collection* and *community*. DSpace search, therefore, is targeted towards retrieval of items only, i.e. search results are always a list of items or collection and community names.

In Figure 3, we notice that the item 1987/117 has a `dcterms:type` 'Book'. Clicking on 'Book' we now see detailed information regarding 'Book' as an *entity* itself. We also notice that we have *indirectly* retrieved every item that has type 'Book' (through the inverse `dcterms:type` property). In addition, we find out that 'Book' belongs to the `dspace-ont:dspacetype` class, clicking on which we trigger semantic search to fetch all instances of this class, which of course are not DSpace items. The same naturally holds for other indirect DSpace entities we have reified, such as authors, formats etc.

5.2 Improved Search Results and Knowledge Discovery

Another advantage of semantic search is to allow retrieval of *more* as well as *more precise* results. These results may be implied by the current data model, but there is no way to retrieve them using the standard configuration.

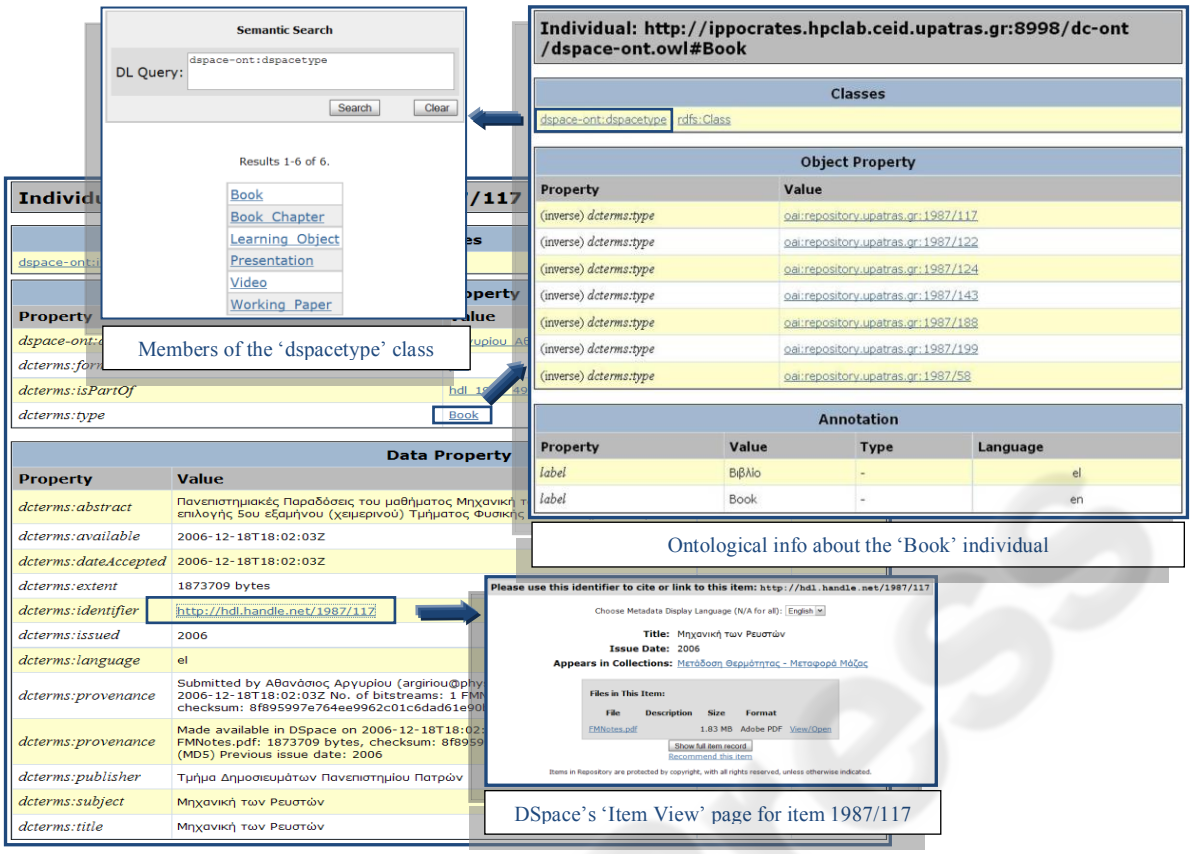

Figure 3: Individual view and navigation pane for item 1987/117.

As an example, suppose we would like to retrieve all items that contain image files. Searching with the keyword 'image' in the traditional repository search returns 2 items. However, examining each of these items metadata reveals that only the latter has actually an 'image/gif' format; the other has a format of 'application/pdf'. The reason why it is returned by traditional search is that DSpace searches also inside the document's text and happens to meet the word 'image'.

On the other hand, the corresponding query through the semantic search interface (Query 1 in Table 1) fetches just one item, exactly the one that has format 'gif'. In this sense, a better level of query *precision* could be sought, since more precise and semantically accurate results can now be obtained.

Suppose now we would like to find out who draws sponsorship from a specific institution, for example, what is funded by the 'Hellenic Ministry of Culture' (Query 2). Searching with these keywords through traditional search returns an item that includes this organization's name in its 'sponsorship' metadata field.

Semantic search however retrieves also the author of this item, aside from the item itself. This is a direct consequence of a role-chain we have declared in our ontology, conveying the fact that authors of items are also receiving sponsorship from the same institution. This example then suggests how semantic search could also improve the *recall* of retrieval, by obtaining a greater number of results.

Table 1: Example queries using the semantic search interface.

	Query (in *Manchester Syntax*)	**Ask for:**
1	`dcterms:format` **`some`** `dspace-ont:image`	*Items that contain image files*
2	`dspace-ont:sponsorship` **`value`** `dspace-ont: Hellenic_Ministry_of_Culture`	*Items/authors that draw sponsorship from a specific institution*
3	**`inv`**`(dspace-ont:author)` **`some`** `(dcterms:format` **`min`** `2 owl:Thing)`	*Authors of items that have at least two different formats*
4	`dspace-ont:co_author` **`some`** `(foaf:name` **`value`** `"Bekiari")`	*The co-authors of an author*

In the 'image' example above, semantic search is able to fetch the particular item, despite the fact that its format is declared just as 'gif' (i.e. it does not contain the keyword 'image'). This is because in our

ontology, 'gif' is an instance of the 'image' class and thus the underlying reasoner is able to conduct an *inference*; that is, since we ask for an 'image' format, we also ask for every instance of this class.

This knowledge discovery capability can also be determined by asking, for example, for the authors of those items (Query 3) that have at least two different formats, using a cardinality restriction on `dcterms:format`. It is easy to see that such a query is impossible to be expressed through traditional search. Similarly, with Query 4 we ask for the co-authors of an author, based on her surname. Due to the definition of the `co_author` property as a role-chain, this request becomes possible and the result is straightforward.

6 CONCLUSIONS

In this paper we have shown how to augment traditional digital repository services by implementing an extensible semantic search and navigation facility on top of DSpace. This facility relies purposely on the OWL API and is designed to be independent of the underlying system, following a "plug-in" philosophy. In combination with the ontology creation and population process, this facility could semantically enable any web-based digital repository system.

Our results confirm that it is possible to navigate among a repository's metadata in more flexible and associative ways. In addition, semantic search can improve traditional keyword search by retrieving more items, but also by fetching more semantically accurate results. And of course, semantic search allows the expression of queries that cannot be expressed by simple keyword-based retrieval.

Finally, it can be seen that the use of ontologies in digital repositories and other information systems, in the way it is suggested in this paper, can benefit from an ontology harvesting and exchange protocol (just as OAI does for metadata), as well as from a standard and semantics-aware language for querying OWL documents.

REFERENCES

Grau, B.C. et al., 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4), pp. 309-322.

Hendler, J., 2008. Web 3.0: Chicken Farms on the Semantic Web. *Computer*, 41(1), pp. 106-108.

Horridge, M., Bechhofer, S. & Noppens, O., 2007. Igniting the OWL 1.1 Touch Paper: The OWL API. In *OWLED 2007, 4th International Workshop on OWL Experiences and Directions*.

Horridge, M. & Patel-Schneider, P., 2008. Manchester Syntax for OWL 1.1. In *OWLED 2008, 5th Int. Workshop on OWL Experiences and Directions*.

Horrocks, I, Kutz, O. & Sattler, U., 2006. The Even More Irresistible SROIQ. *In KR2006, 10th Int. Conf. on Principles of Knowledge Representation and Reasoning*.

Horrocks, I., 2008. Ontologies and the Semantic Web. *Communications of the ACM*, 51(12), pp. 58-67.

Koutsomitropoulos, D., Meidanis, D., Kandili, A. & Papatheodorou, T., 2008a. Establishing the Semantic Web Reasoning Infrastructure on Description Logic Inference Engines. In Y. Manolopoulos et al., eds. *Enterprise Information Systems, Lecture Notes in Business Information Processing*. Springer, pp. 351-362.

Koutsomitropoulos, D., Solomou, G. & Papatheodorou, T., 2008b. Semantic Interoperability of Dublin Core Metadata in Digital Repositories. In *Innovations'08, 5th Int. Conf. on Innovations in Information Technology*.

Koutsomitropoulos, D., Paloukis, G. & Papatheodorou, T., 2009a. Semantic Application Profiles: A Means to Enhance Knowledge Discovery in Domain Metadata Models. In M.A. Sicilia and M. Lytras, eds. *Metadata and Semantics*. Springer, pp. 23-34.

Koutsomitropoulos, D., Solomou, G., Alexopoulos, A. & Papatheodorou, T., 2009b (in press). Semantic Metadata Interoperability and Inference-Based Querying in Digital Repositories. *Journal of Information Technology Research*. (Accepted for publication March 2009).

Kruk, S.R., Decker, S. & Zieborak, L., 2005. JeromeDL - Reconnecting Digital Libraries and the Semantic Web. In *WWW2005, 14th Int. World Wide Web Conference.*

Lagoze, C., Van de Sompel, H., Nelson, M. & Warner, S., 2002. The Open Archive Initiative Protocol for Metadata Harvesting. [Online] Available at: http://www.openarchives.org/OAI/openarchivesprotocol.html [Accessed December 2008].

Nilsson, M, Powell, A., Johnston, P. & Naeve, A., 2008. Expressing Dublin Core metadata using the Resource Description Framework (RDF). DCMI Recommendation. [Online] Available at: http://dublincore.org/documents/dc-rdf/ [Accessed January 2009]

Nucci, M., Hahn, D. & Barbera, M., 2008. The Talia Library Platform - Rapidly Building a Digital Library on Rails. In *SFSW2008, 4th Workshop on Scripting for the Semantic Web, European Semantic Web Conference*.

Risse, T., Knezevic, P., Meghini, C., Hecht, R. & Basile, F., 2005. The BRICKS Infrastructure - An Overview. In *EVA 2005, 8th Annual Int. Conf. on Electronic Information and Visual Arts*.

Sirin, E. & Parsia, B., 2007. SPARQL-DL: SPARQL Query for OWL-DL. In *OWLED 2007, 4th Int. Workshop on OWL Experiences and Directions*.