Semantic Interoperability of Dublin Core Metadata in Digital Repositories

Dimitrios A. Koutsomitropoulos, Georgia D. Solomou, Theodore S. Papatheodorou, *member IEEE*

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

{kotsomit, solomou, tsp}@hpclab.ceid.upatras.gr

Abstract

Metadata applications have evolved in time into highly structured "islands of information" about digital resources, often bearing a strong semantic interpretation. Scarcely however are these semantics being communicated in machine readable and understandable ways. At the same time, the process for transforming the implied metadata knowledge into explicit Semantic Web descriptions can be problematic and is not always evident. In this paper we take upon the well-established Dublin Core metadata standard and suggest a proper Semantic Web OWL ontology, coping with discrepancies and incompatibilities, indicative of such attempts, in novel ways. Moreover, we show the potential and necessity of this approach by demonstrating inferences on the resulting ontology, instantiated with actual Dublin Core metadata, originating from the live DSpace installation of the University of Patras institutional repository.

1. Introduction

In many standard configurations (including the DSpace digital repository software) the DC Metadata Element Set (DCMES) is implemented as a flat aggregation of elements. This is also true for qualifiers, which are not always implemented as sub-properties of main elements; rather, they often appear at the same level as parent elements and the sub-element/qualifier relationship is maintained only in the label. This situation, evident also in the DSpace-based University of Patras institutional repository is depicted in figure 1.

The semantic interpretation of the DC model that, as we see, is not always represented in applications, is formalized through the DCMI Abstract Model specification [11] as well as the most recent recommendation for expressing DC in RDF [9]. These documents virtually suggest an ontology of DC, expressed in RDF(S), a Semantic Web standard.

Such a DC ontology bears its own semantic structure that may be taken advantage of in order to enable more refined descriptions of resources. This of course is reminiscent of the well-known Semantic Web "bootstrapping problem" (e.g. see [4], [5]): The availability of high-quality, complex and interconnected resource descriptions is a key aspect for the Semantic Web to be of some value; on the other hand, the burden to create a whole new set of rich annotations is too high, both from a conceptual (hard to conceive) as well as from an effort (too much time) point of view.

Having these in mind, we propose an implementation of the DC ontology that is to be carried out in terms of a most centralized approach. To do this we are based on the *semantic profiling* technique, well-applied previously on fully-structured knowledge domains, such as the CIDOC-CRM [1] and introduced in [6]. Using this technique we try to better capture the intended semantics of the DC metadata domain, having the DC RDF(S) schema as a starting point. Our goal is to upgrade this ontology up to OWL and OWL 1.1 (now OWL 2) level [10], by incorporating new constructs and refinements, available only in these languages. At the same time, we build upon the initial model and do not require any alternations in its original specification. The resulting ontology, including the new refinements, is then populated in an automated way from DC metadata already existing within the live DSpace installation of the University of Patras institutional repository (http://repository.upatras.gr/dspace/), using the system's OAI interface (see also section 3).

DC Field	Value	Language
dc.contributor.author	Κουτσομητρόπουλος, Δημήτριος	el
dc.contributor.author	Koutsomitropoulos, Dimitrios	en
dc.date.accessioned	2006-12-15T13:13:49Z	-
dc.date.available	2006-12-15T13:13:49Z	-
dc.date.issued	2006-12-15T13:13:49Z	-
dc.identifier.uri	http://hdl.handle.net/1987/104	-
dc.description	Εργαστήριο Πληροφοριακών Συστημάτων Υψηλών Επιδώσεων	el
dc.description	HPCLab	en
dc.description.abstract	Στην παρούσα παρουσίαση περιγράφονται ενέργειες προετοιμασίας πρωτοτύπων που στόχο έχουν τη διατήρηση και τη διάσωση της ποιότητας του περιεχομένου.Παράμετροι που αναφέρονται και αναλύονται είναι η καταλληλότητα του εξοπλισμού και των συνθηκών περιβάλλοντος και η μεταχείριση πρωτοτύπων.	el
dc.description.abstract	The presentation above describes preparation actions that intent to the preservation and the rescue of the content quality.Factors like equipment and environment appropriateness as well as prototype usage are analytically adverted.	en

Figure 1. Detailed item view in DSpace

The rest of this paper is organized as follows: In section 2 we describe our process for creating a DC web ontology: we discuss options for coping with identified problems, present our implementation steps and provide indicative examples of our elaborations. Following, section 3 outlines the experimental configuration for the automatic population of the ontology with real data. It then comments on the capabilities enabled by presenting a couple of reasoning-based queries, conducted through the Protégé environment. Finally, section 4 summarizes our conclusions and future work.

2. Creating a Dublin Core OWL Ontology

In order to implement our OWL DC ontology we take as a starting point the DCMI recommended RDF implementation. We then create a semantic application profile for our actual metadata repository, following the technique presented in [6].

An important variation however is that here, we do not have a well-structured knowledge domain as the one expressed by CIDOC-CRM, but our instances are virtually a flat aggregation of elements. The DC RDF implementation may be a good starting point, but there is still a gap to be filled in order to reach this state. Therefore, we need a transition from an absent semantic model to an item-property-predicate situation, thus invoking the RDF semantics. This transition is offered by an XSLT (see next section for details) which can be considered part of the syntax transformation stage in the semantic profiling process.

Implementation steps can be organized in accordance to [6], in three stages, namely *Syntax Transformation*, *Semantic Intension* and *Semantic Refinement* (see also Figure 2):

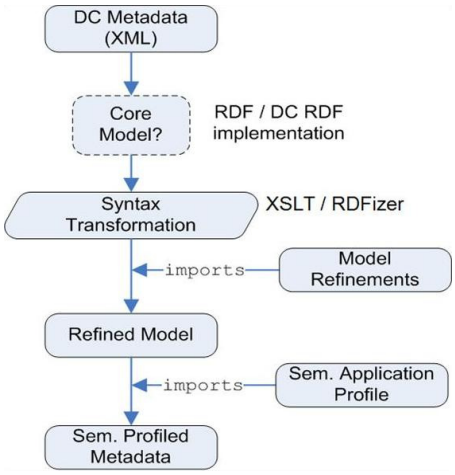

Figure 2. Implementation steps of the DC ontology

2.1. Syntax Transformation

This transformation amounts to:

a) Minimal syntax transformation to "clean-out" OAI overhead statements.

b) Assign types (datatypes) to literal values. DC offers an abstraction for datatypes, called *syntax encoding schemes*. These schemes are in fact equivalent to the actual XML Schema datatypes that are more likely to be supported in OWL applications, since in OWL 2 it is specified that an application may signal an error if an unsupported datatype is encountered [8]. For example, `dcterms:W3CDTF` corresponds to `xsd:date`, `dcterms:RFC3066` to `xsd:language` and so on.

c) Reify literals (as objects) originating from specific (controlled) vocabularies, such as MIME types. In particular, DSpace relates items with literals corresponding to MIME types through the

dcterms:format property. The use of MIME types as fillers to dcterms:format is also a DCMI recommended practice [3].

d) Reify other literals to be able to identify and express potential semantic relations. For example, DSpace author names are defined as objects and their comma separated first and last names (as stored by DSpace) are parsed and assigned to corresponding properties from the popular FOAF ontology (http://xmlns.com/foaf/spec/).

e) Introduce and re-assign namespaces. This means that dc: is replaced by dcterms: and also dspace-ont: for DSpace specific elements is introduced.

The result of this stage is the creation of an RDF document that contains triples describing the repository resources, mixed with the XML schema and FOAF namespaces; let it be 'instances.rdf'.

2.2. Semantic Intension

Second, we can commence with the semantic intension of the DC model. To do this we need to establish (or devise) a core model first. Such a model is

- offered by the RDF semantics, inherent in the DC RDF implementation,
- further profiled by the new DC implementation that incorporates DCAM as well as domain and range restrictions,
- further profiled by post-RDF characterizations of properties, like inversity, symmetry, transitivity, functionality. For example, we define dcterms:relation as symmetric and dcterms:hasPart as the inverse of dcterms:isPartOf.

The result of this stage is the creation of an OWL document that imports the original dcterms RDF implementation document ('dcterms.rdf') and contains the above refinements (e.g. 'dc-ont.owl'). This document comprises the DC ontology, expressed in OWL format, which refines the original specification by utilizing OWL-specific constructs, but retaining at the same time its interoperability.

2.3. Semantic Refinement

Now, it is time to add semantic refinements for our particular application scenario, i.e. the University of Patras institutional repository. This is conducted by:

a) Model vocabularies in subsumption hierarchies. For example, the MIME type vocabulary that DSpace supports is implemented as a partition of subclasses, with instances, and is related to dcterms:FileFormat.

b) Identify and represent DSpace notions of 'item', 'collection' and 'community' as classes. 'Collection' is further defined as a subclass of 'community', conveying the fact that, in DSpace, collections refine communities. Items relate to collections through the dcterms:isPartOf property.

c) Identify and represent the non-DC notions of 'author' and 'sponsorship' and relate them to the initial model: we define, under a different namespace, author and sponsorship as OWL object properties, and make them sub-properties of dcterms:contributor and dcterms:description respectively.

d) Model complex relations using role-forming operators and restrictions ('some'- ∃ and 'for all' - ∀): For example, we define the notion of "co-author" as

$$\text{author}^{-} \circ \text{author} \sqsubseteq \text{co\_author},$$

and refine sponsorship by

$$\text{dcterms:contributor}^{-} \circ \text{sponsorship} \sqsubseteq \text{sponsorship},$$

meaning that authors of items are also receiving sponsorship from the same institution. We also state that:

$$\neg\text{item} \sqcap \exists\text{dcterms:hasPart.item} \sqsubseteq \text{collection},$$

meaning that everything that 'hasPart' an item is a collection, unless it is an item itself. In addition, collections may only have items as parts:

$$\text{collection} \sqsubseteq \forall\text{dcterms:hasPart.item}.$$

This stage results in creating a new OWL 2 document ('dspace-ont.owl') containing OWL constructs that refine the DSpace data model in a semantic manner.

The implementation steps described above are outlined in figure 2. All documents are available at: http://ippocrates.hpclab.ceid.upatras.gr:8998/dc-ont.

3. Results

3.1. Technical Configuration

In order to apply and evaluate our technique we instantiate our ontology with descriptions coming from the University of Patras institutional repository which is based on DSpace. DSpace offers the ability

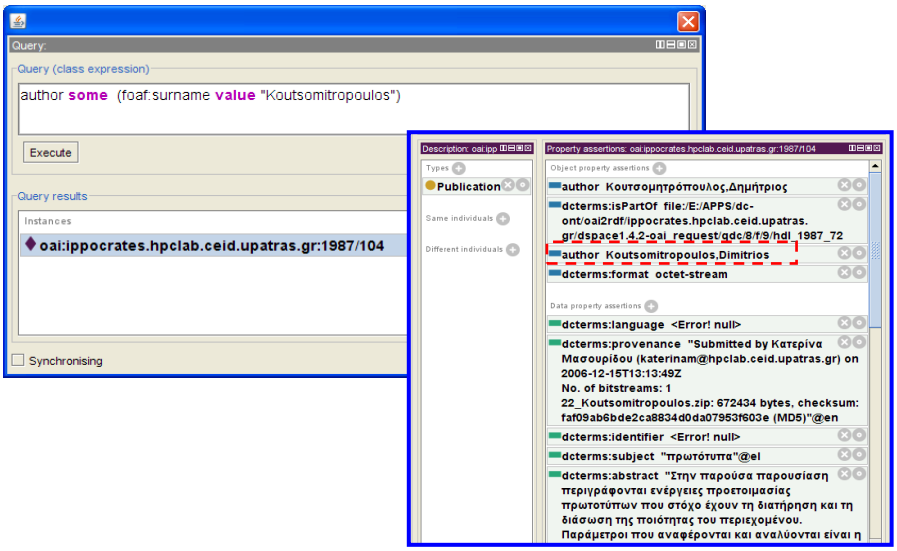

Figure 3. String-based query and result through Protégé 4.0

to harvest its metadata by providing an OAI-PMH interface. Internally it follows a particular application profile, borrowing heavily from the Library Application Profile [2]. Including qualifications, DSpace fosters a total of 66 elements, not all of which are visible to the user.

OAI-PMH is an HTTP based protocol for interoperable metadata harvesting [7], and OAI-compliant harvesting interfaces and service providers are becoming common in digital repositories and libraries. Though it would be easier to access the repository's database directly, we opted for OAI, precisely in order to show in practice how our approach can be generalized.

The repository's OAI-PMH facility is configurable as to what elements are to be exported (including their namespace and label) and also supports simple DC (oai_dc) as well as its qualifications (qdc). We opt for the latter as richer from a semantic point of view and employ the RDFizer tool (http://simile.mit.edu/wiki/OAI-PMH_RDFizer), developed under the Simile project, that actually harvests OAI metadata and saves them in RDF format. In this process however we devise our own XSLT stylesheet that tries to better capture the semantic relations implied in the metadata as well as to construct the OWL specific instantiations.

The semantic application profile, saved in dspace-ont.owl, is then imported in the transformed RDF document. So does the dcterms.rdf document. The resulting ontology is then manipulated through the Protégé 4 environment.

3.2. Conducting Queries

In the following we attempt to pose a series of intelligent queries to the produced OWL (2) document. To do this, we use the DL Query Tab of Protégé 4, testing both its bundled reasoners, FaCT++ and Pellet.

First we conduct a string-based query that would also be possible from inside the standard DSpace querying interface. In the query shown in figure 3, we ask for the items authored by an author having a particular surname. By conducting some similar inferences including conjunctive ones, we confirm that it is possible to produce identical results and conclude that our ontology is at least semantically equivalent to the existing model.

Second, we carry out some inferences that are based on the new semantic constructs that have been added to the profile. The results may be implied by the current data model, but there is no way to retrieve them using the standard configuration. For example we can ask for the co-authors of a particular author, or where a particular author draws sponsorship from. In figure 4 we ask "who has authored an item comprising of more than one file types?" using a cardinality restriction on `dcterms:format`. Since item 1987/122 is related to two distinct MIME types, namely gif and pdf, the result follows.

We can also perform queries involving the other semantic refinements we have introduced. The results lead us to the conclusion that the proposed approach enables better information discovery with little manual intervention.

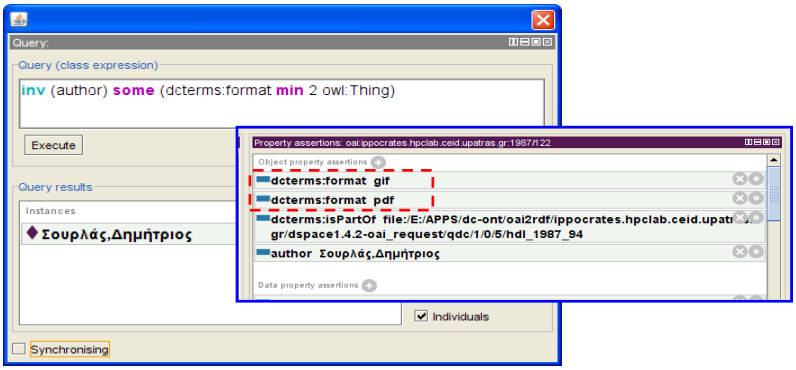

Figure 4. DL-like query and result through Protégé 4.0 using cardinality restrictions

4. Conclusions and Future Work

Throughout this work, we have shown how the alignment of DC to the Semantic Web is crucial for its future interoperability and utilization. The application of the semantic profiling technique in a domain with little semantic structure as the one underlying flat DC metadata implementations (in addition to fully semantically structured domains as the CIDOC-CRM) can also produce added value in terms of knowledge discovery and semantic interoperability. The latter is achieved exactly by "unleashing" the repository's metadata in web-accessible OWL format, but most importantly, by the "upgrade" of this metadata in concrete semantic structures (reification).

Our efforts point towards a direction that achieves semantic enrichment of existing flatly described resources in a centralized manner; in such a way the burden on content curators and end-users is alleviated and a potential solution to the Semantic Web "chicken-egg" problem [5] is suggested.

We are also working on extending the ontological profile with learning object metadata (LOM) and investigating semantic relationships with and mapping to DC-based information, in this ontological context. Early results confirm that new metadata concepts can be seamlessly integrated, a fact that owes to the design of the semantic profiling technique and verifies the interoperability of our approach.

Acknowledgments

Part of this work has been conducted under the University of Patras "Operational Program for Education and Initial Vocational Training" (ΕΠΕΑΕΚ ΙΙ), funded by the Hellenic Ministry of Education and the European Commission.

References

[1] N. Crofts, M. Doer, and T. Gill, "The CIDOC Conceptual Reference Model: A standard for communicating cultural contents", *Cultivate Interactive*, Issue 9, 2003, http://www.cultivate-int.org/issue9/chios/

[2] DCMI-Libraries Working Group: Library Application Profile, 2004, http://dublincore.org/documents/2004/09/10/library-application-profile/

[3] DCMI Usage Board: DCMI Metadata Terms, Jan. 2008, http://dublincore.org/documents/dcmi-terms/

[4] S. Dill, D. Eiron, D. Gibson, et al., "SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation", *Proc. 12th International Conference on World Wide Web*, 2003, pp. 178–186.

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

[6] D. Koutsomitropoulos, G. Paloukis, and T. Papatheodorou, "From Metadata Application Profiles to Semantic Profiling: Ontology Refinement and Profiling to Strengthen Inference-based Queries on the Semantic Web", *Int. J. on Metadata, Semantics and Ontologies*, 2 (4), 2007, pp. 268-280.

[7] C. Lagoze, H. Van de Sompel, M. Nelson, and S. Warner, The Open Archive Initiative Protocol for Metadata Harvesting, 2002, http://www.openarchives.org/OAI/openarchivesprotocol.html

[8] B. Motik, and I. Horrocks, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax, 2008, http://www.w3.org/TR/2008/WD-owl2-syntax-20080411/

[9] M. Nilsson, A. Powell, P. Johnston, and A. Naeve, Expressing Dublin Core metadata using the Resource Description Framework (RDF), 2008, http://dublincore.org/documents/dc-rdf/

[10] P.F. Patel-Schneider, and I Horrocks, OWL 1.1 Web Ontology Language Overview, 2007, http://www.webont.org/owl/1.1/overview.html

[11] A. Powell, M. Nilsson, A. Naeve, P. Johnston, and T. Baker, DCMI Abstract Model, 2007, http://dublincore.org/documents/abstract-model/