Semantic Application Profiles: A Means to Enhance Knowledge Discovery in Domain Metadata Models

Dimitrios A. Koutsomitropoulos, George E. Paloukis and Theodore S. Papatheodorou

Abstract. Ontologies on the Semantic Web form a basis for representing human-conceivable knowledge in a machine-understandable manner. Ontology development for a specific knowledge domain is however a difficult task, because the produced representation has to be adequately detailed and broad enough at the same time. The CIDOC-CRM is such an ontology, pertaining to cultural heritage, which we align to the Semantic Web environment: first transforming it to OWL and then profiling it not in the usual flat metadata sense, but by refining and extending its conceptual structures, taking advantage of OWL semantics. This kind of profiling maintains applicability of the model, while enabling more expressive reasoning tasks. To this end, we construct a mechanism for acquiring implied and web-distributed information that is used to conduct and present a series of experimental inferences on the CRM profiled form.

1 Introduction

The CIDOC Conceptual Reference Model [1] is an ontology that attempts to model the knowledge domain of cultural heritage. As every human conceivable domain, cultural heritage is very hard to be accurately modeled. In addition and due to its nature, cultural heritage information use to be hidden in libraries and museum archives, and when available on-line are poorly or not at all structured. Moreover, the CIDOC-CRM has been recently appointed an ISO standard status (ISO-21127), a fact that further stresses its importance as a common conceptual basis between cultural heritage applications.

High Performance Information Systems Laboratory,
School of Engineering, University of Patras,
Buidling B, 26500 Patras-Rio, Greece
{kotsomit, gpalouk, tsp}@hpclab.ceid.upatras.gr

Using the CIDOC-CRM standard as our conceptual basis, we first create its machine meaningful counterpart by expressing it in the Web Ontology Language (OWL), a W3C standard. This process does not merely amount to a simple syntax transformation. Rather, taking advantage of OWL most expressive (but, simultaneously, decidable) structures we also enrich and upgrade the model, thus further narrowing the conceptual approximation [4].

To avoid making the model too specific, we incorporate the OWL statements in different documents that involve concrete instances of the CRM's concepts and roles. This approach not only demonstrates the distributed knowledge discovery capabilities inherent in web ontologies; at the same time it suggests a *semantically enhanced application profiling* paradigm: this kind of profiling takes the usual metadata-specific sense, where it is seen as an aggregation of disparate metadata elements [5], a step further: It does not so much deal with a horizontal extension of the ontology, but rather extends it in a semantic manner, as may dictated by a particular application.

The next step is to take advantage of this new ontology mostly by being able to reap the benefits of our semantic extensions. Thus we employ a methodology [6] and implement a prototype web application, the Knowledge Discovery Interface (KDI) to be able to pose reasoning-based intelligent queries to the CRM profiled form.

The rest of this paper is mainly organized as follows: First we give an overview of the metadata application profiling idea and approaches. Then we present our process of transforming and profiling the CIDOC-CRM, pointing out our extensions and discussing semantic profiling; following there are the inferences conducted on the CRM using the KDI and their results. Finally, we summarize potential future work and the conclusions drawn from our approach.

2 Metadata Application Profiling

The need for efficient resource description in electronic archives quickly identified the lack of uniform ways for representing and maintaining information about resources. These pieces of information, known as *metadata* would therefore be organized in concrete schemata produced and managed by content authorities, institutions and domain experts. The XML language eased this process by providing an official syntax for expressing both schemata and actual metadata information in machine readable format.

However, as these schemata tended to proliferate day by day, focusing on a particular domain of interest or function, there was often the case where particular developer's needs were not satisfied by any existing schema or some elements he may find suitable where scattered over various standard implementations. Metadata application profiling came then as a natural means to overcome these obstacles while respecting the standards *raison d'être*: As defined in [3] application profiling is the assemblage of metadata elements selected from one or more metadata schemas and their combination in a compound schema.

Let's briefly examine an application profiling example [9]:

```
<?xml version="1.0"?>
<record
 xmlns="http://example.org/learningapp/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-
 instance"
 xsi:schemaLocation="http://example.org/learningapp/
 schema.xsd"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ims="http://www.imsglobal.org/xsd/imsmd_v1p2">
  <dc:title>
    Frog maths
  </dc:title>
  <dc:description>
    Simple maths games for 5-7 year olds.
  </dc:description>
  <ims:typicallearningtime>
    <ims:datetime>
      0000-00-00T00:15
    </ims:datetime>
  </ims:typicallearningtime>
</record>
```

This is an instantiation of mixing Dublin Core elements with IMS learning metadata. It is noticeable that the most important means for actually implementing an application profile are *namespaces*. In the example above `dc` represents elements from the Dublin Core set, while `ims` denotes IMS-originating metadata. Namespaces play a crucial role not only in identifying provenance of distinct schemata, but also as a means to separate and then merge different elements and vocabularies.

It is clear that metadata schemata attempt to capture and convey human conceivable knowledge in the most basic unambiguous machine-compatible form: A horizontal aggregation of definitions (possibly with sub-elements) with specified value restrictions and formats that is expressed (most often) in XML. Metadata standards are perfectly successful in this manner; at the same time their representation of knowledge is considered quite poor and distanced from machine-understandability.

On the other hand, the Semantic Web and its ontologies give the chance of more accurate modeling of domain knowledge: ontologies are essentially metadata schemata with precisely defined meaning and richer relations between elements and concepts of a conceptual model. With this new toolbox at hand a series of possibilities is now opened that may further ease the development of enhanced metadata profiles. These include a novel method for creating a metadata application profile, not just by combining, refining or restricting elements, but by the semantic extension of the model, and that is exactly what we are trying to do in the following section.

3 Transforming the CIDOC-CRM

CIDOC-CRM includes about 84 concepts and 139 roles, not counting their inverses (that is, a total of 278 roles) (Figure 1). In terms of expressivity, the CRM employs structures enabled by RDF(S), which may be summarized as follows:

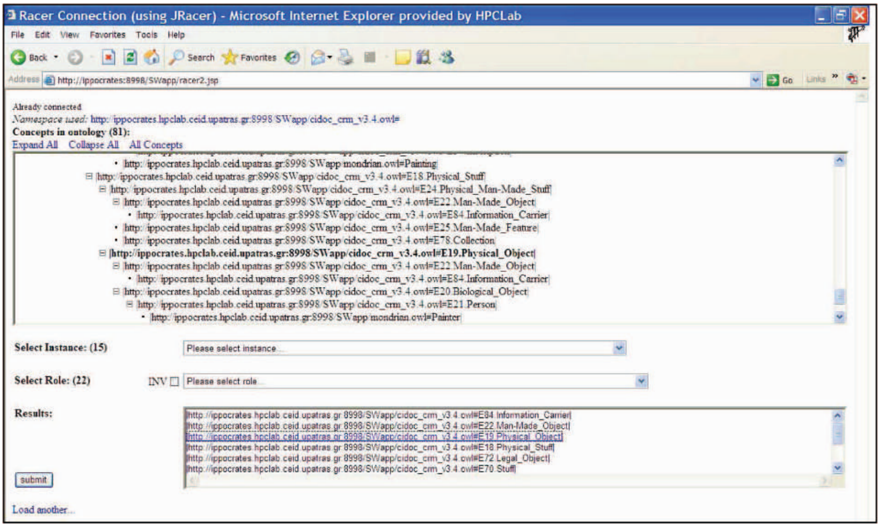


Fig. 1. CIDOC-CRM taxonomy as shown by the KDI.

- Concepts as well as roles are organized in hierarchies.
- For every role, concepts are defined that form its domain and its range.
- For every role, its inverse is also defined, as a separate role, because RDF(S) cannot implicitly express inversion relation between two roles.
- There is no distinction between object and datatype properties (roles) as in OWL; Rather, roles that are equivalent to datatype properties have rdf:Literal as their range.

As of Jan. 2005 there exists an OWL transcription of the CRM's RDF document. However this version adds only role specific constructs (inversion, transitivity etc) which, semantically, do not exceed OWL Lite.

To create a CIDOC-CRM semantic application profile, we follow a twofold approach: First we transcode it in an appropriate and expressive format (namely OWL); Next, we commence with its actual profiling, first by strengthening its intension, i.e. the general knowledge about the domain [8] and then by refining the model for the needs of a particular application.

Although the CRM syntactic processing is not straightforward, in the following we are focusing on the semantic transformation of the model, the details of the transcoding being presented in [7].

3.1 Semantic Intension and Refinement

This phase of CRM upgrading includes its semantic augmentation with OWL-specific structures up to the OWL DL level, so as to enable a satisfactory level of reasoning, as well as its completion with some concrete instances.

This has been conducted in two steps: first, we added expressions that pertain to the model itself, so as to better capture intended meaning of properties and classes by taking advantage of OWL vocabulary. Second, added further subclasses and *semantic constraints* on them that actually profile the model for the specific case of paintings and painters in general. As an application scenario we have chosen to model facts from the life and work of the Dutch painter Piet Mondrian. Let's examine these steps in detail:

Core Intension Strengthening

In this step, we do not add any new classes or entities that extend the CRM. Instead, we try to better approximate the core model's conceptualization by using OWL statements that allow for its more precise implementation. In particular:

- We modelled minimum and maximum cardinality restrictions by using unqualified number restrictions (owl:minCardinality, owl:maxCardinality).
- We modelled inverse roles, using the owl:inverseOf operand.
- We included a symmetric role example, using the rdf:type= "&owl;Symmetric" statement.

Note that RDF(S) being CRM's favoured implementation, there is no way to express such constraints. For the purpose of our work, we have not exhaustively quantified the CRM properties, but applied constraints to some ones, used and instantiated in our Mondrian example (see Section 4).

Clearly, the additions above actually refine the core model in a way that achieves to expand the *intensional knowledge* of the schema using constructs and means provided only in a Semantic Web infrastructure.

Application Refinement

During this step, we create some specific CRM concept and role instances pertaining to our particular application. We also include axiom and fact declarations that only OWL allows to be expressed, as well as new roles and concepts making use of this expressiveness.

- We added the classes: “Painting” as subclass of CRM’s “Visual_Item”, “Painting_Event”, a subclass of “Creation_Event” and “Painter” a subclass of “Person”.
- We added a data type property “hasURL” as a sub-property of “has_current_location”.
- We semantically characterized above concepts based on existential and universal quantification, by using the owl:hasValue, owl:someValuesFrom and owl:allValuesFrom expressions, which ultimately enable more complex inferences.

This is another direction of semantic profiling: We added new elements bearing their own namespace, but then we semantically entangled them with each other and with the model’s own definitions, thus imposing *semantic refinements* for our own specific case.

3.2 A Semantic Profiling Technique

The above discussion introduces the process of creating semantic application profiles and suggests a universal paradigm for Semantic Web metadata applications. Although we applied this technique specifically on CIDOC-CRM, it can be easily seen that it fits any other domain of interest. Independently of the domain chosen, one has first to consider a suitable machine readable implementation for the model, which, for the time being, is offered by the OWL specifications.

Given a proper syntax, it is worth examining the possibilities of better capturing the intensional knowledge of the model, taking advantage of any particular vocabulary the representation language may offer. In this way the conceptualization of the model is strengthened and its potential ensured.

At some point the initial model may be found inadequate for the specific application needs. As is the case with traditional metadata profiling, other ontological and metadata schemata may have to be considered and mixed with the original, thus revisiting the initial step. In addition, one can devise appropriate constructs in order to narrow the semantics of the intended application.

One of the main concerns when developing an application profile is to ensure the source schema is not affected and its general applicability maintained. To achieve this, in addition to namespaces, OWL provides an explicit inclusion mechanism through the `<owl:imports>` statement. In our case, we chose to include our semantic ornaments in three new OWL documents, namely: crm_core_profile.owl for the core intension, crm_paint_profile.owl for the application refinement and mondrian.owl as the instantiation of the above[2]. In this

[2]All documents are available under http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/

way we preserve the original model and we also show Semantic Web capabilities for ontology integration and distributed knowledge discovery.

4 Results

In the following we present the results from some experimental inference actions conducted on the CRM augmented OWL form using our KDI. The KDI is a web application, providing intelligent query submission services on Web ontology documents and is detailed in [6].

Inferences performed can be divided in two categories: *Positive* inferences where, based on the concept and role axioms as well as the ontology facts we conclude new, not explicitly expressed facts and *negative* inferences where, based on the ontology axioms and facts we detect unsatisfiability conditions on concepts and instances.

4.1 Positive Inferences

The following code is a fragment from mondrian.owl stating that a “Painting_Event” is in fact a “Creation_Event” that “has_created” “Painting” objects only:

```
<owl:Class rdf:ID="Painting_Event">
  <rdfs:subClassOf
  rdf:resource="&crm;E65.Creation_Event"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
      rdf:resource="&crm;P94F.has_created"/>
      <owl:allValuesFrom rdf:resource="#Painting"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<Painting_Event rdf:ID="Creation of Mondrian's
composition">
```

Top Concept: **T**

P94F.has_created: **R**

Painting_Event: **C**

Painting: **D**

Creation of Mondrian’s Composition: $\mathbf{i_1}$

Mondrian’s Composition: $\mathbf{i_2}$

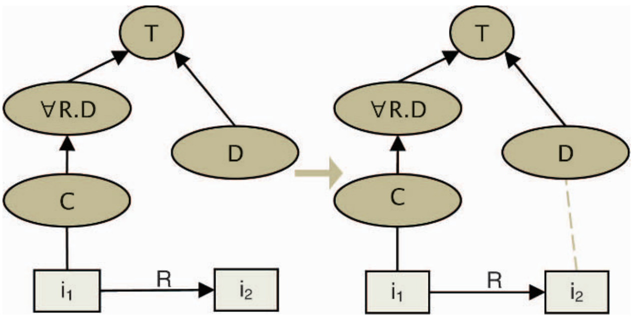


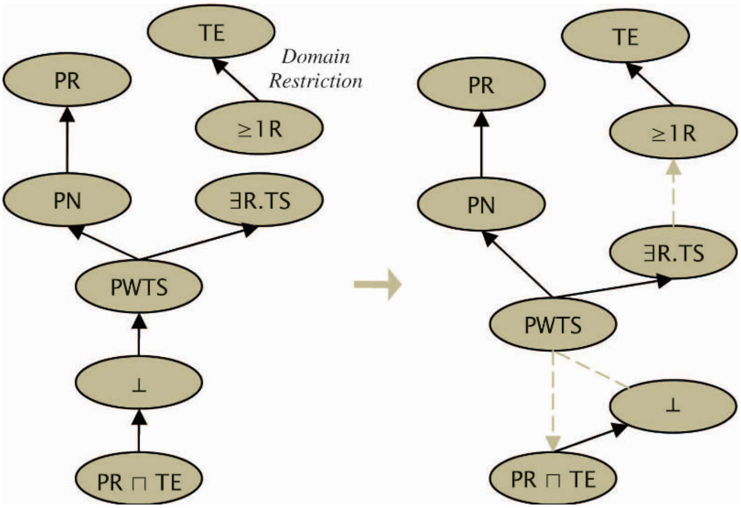
Fig. 2. Inference example using value restriction.

```
<crm:P94F.has_created rdf:resource="#Mondrian's
  composition"/>
</Painting_Event>
```

The above fragment is graphically depicted in the left part of Figure 2. Creation of Mondrian's Composition" (i_1) is an explicitly stated "Painting_Event" that "has_created" (R) "Mondrian's composition" (i_2). Now, asking the KDI to infer "what is a painting?" it infers that i_2 is indeed a painting (right part of Figure 2), correctly interpreting the value restriction on role R.

4.2 Negative Inferences

In CRM, temporal events may have a time-span. Naturally, a "Person" cannot have a time-span, unless it is also a "Temporal Entity". In the following we state that "Persons" and "Temporal Entities" are disjoint concepts and we attempt to define the class of "Painters with time-span".



Bottom Concept: ⊥

P4F.has_time-span: **R**

E2. Temporal_Entity: **TE**

E21.Person: **PR**

E52.Time-Span: **TS**

Painter: **PN**

Painter_with_time-span: **PWTS**

Fig. 3. Detecting unsatisfiable concepts.

```
<owl:ObjectProperty rdf:ID="P4F.has_time-span">
  <rdfs:domain rdf:resource="#E2.Temporal_Entity"/>
</owl:ObjectProperty>
<owl:Class rdf:about="&crm;E2.Temporal_Entity">
  <owl:disjointWith rdf:resource="&crm;E21.Person"/>
</owl:Class>
<owl:Class rdf:about="#Painter">
  <rdfs:subClassOf rdf:resource="&crm;E21.Person"/>
</owl:Class>
<owl:Class rdf:ID="Painter_with_time-span">
  <rdfs:subClassOf rdf:resource="#Painter"/>
  <rdfs:subClassOf>
    <owl:Restriction>
     <owl:onProperty rdf:resource="&crm;P4F.has_time-
      span"/>
      <owl:someValuesFrom rdf:resource="&crm;E52.Time-
      Span"/>
   </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

The above fragment is graphically depicted in Figure 3. A “Painter with time-span” is defined as a “Painter” (known subclass of “Person”) that “has time-span” some “Time-Span” instances. However, individuals that “have time-span” are required to belong to the “Temporal Entity” class, as dictated by the corresponding domain restriction. Therefore, apart from being a “Person”, a “Painter with time-span” must also be a “Temporal Entity”. On the other hand “Persons” and “Temporal Entities” are disjoint, so their intersection represents the bottom (always empty) concept. Thus, a “Painter with Time-Span” can never exist, as its class is inferred to be equivalent to the bottom concept. The KDI correctly detects the unsatisfiability of this class by pointing it out with red colour in the taxonomy.

5 Conclusions and Future Work

In this paper we attempted to deploy a working platform upon which we experimented with the application of Semantic Web techniques and ideas on the cultural heritage domain. Concurrently, we suggested a practice that can easily be followed in any other domain of interest.

First we have shown the Semantic Web capabilities for knowledge discovery with web ontologies. We conducted and presented a series of successful experimental results possible only after aligning our ontological model to the Semantic Web standards. A side product of this process is the strengthening of the argument that OWL and its most expressive decidable subset, OWL DL, may be recommended for modeling domain metadata and be fruitful in that way.

Doing so, we elaborated a novel technique for creating metadata application profiles, by taking advantage of the Semantic Web toolbox. This technique, involves semantic enrichment of the metadata model and then deepening of its structures and definitions in accordance to specific needs. Having the CIDOC-CRM as a starting point, our approach can be likewise applied in any other knowledge domain.

A possible combination of semantic profiling with traditional metadata profiling practices like namespace inclusion and merging may be worth to examine as future work. The combination for example of a CRM profile with a flat metadata schema (e.g. Dublin Core) should allow for the interchangeable use of both their element sets, provided this is done in a semantically consistent and productive manner, i.e. simple metadata elements are not treated naively as annotations.

To this end, of particular interest is looking into the upcoming OWL 1.1 [2] specification and especially its concept of *punning* as a meta-modeling principle, based on which a name definition may have variable semantic interpretation depending on the ontological context.

Acknowledgments. Dimitrios Koutsomitropoulos is partially supported by a grant from the “Alexander S. Onassis” Public Benefit Foundation.

References

1. Crofts N, Doerr M, Gill T (2003) The CIDOC Conceptual Reference Model: A standard for communicating cultural contents. Cultivate Interactive, 9. http://www.cultivate-int.org/issue9/chios/
2. Cuenca Grau B, Horrocks I, Parsia B et al (2006) Next Steps for OWL. In Proc. of the OWL Experiences and Directions Workshop (OWLED’06)
3. Duval E, Hodgins W, Sutton S et al (2002) Metadata Principles and Practicalities. D-Lib Magazine, 8. http://www.dlib.org/ dlib/april02/weibel/04weibel.html
4. Guarino N (1998) Formal Ontology and Information Systems. In N. Guarino (ed.): Formal Ontology in Information Systems. Proc. of FOIS’98. IOS Press
5. Heery R, Patel M (2000) Mixing and Matching Metadata Schemas. Ariadne, 25. http://www.ariadne.ac.uk/issue25/app-profiles/intro.html
6. Koutsomitropoulos DA, Fragakis MF, Papatheodorou TS (2006) Discovering Knowledge in Web Ontologies: A Methodology and Prototype Implementation. In Proc. of SEMANTICS 2006. OCG 151-164
7. Koutsomitropoulos DA, Papatheodorou TS (2007) Expressive Reasoning about Cultural Heritage Knowledge Using Web Ontologies. In Proc. of 3d International Conference on Web Information Systems and Technologies (WEBIST 2007). WIA track, pp.276-281

8. Nardi D, Brachman RJ (2007) An Introduction to Description Logics. In Baader F, Calvanese D, McGuiness DL et al (eds) The Description Logic Handbook, 2^{nd} edn. Cambridge
9. Powell A, Johnston P (2003) Guidelines for implementing Dublin Core in XML. DCMI Recommendation. http://dublincore.org/documents/dc-xml-guidelines/