From metadata application profiles to semantic profiling: ontology refinement and profiling to strengthen inference-based queries on the semantic web

Dimitrios Koutsomitropoulos*, George Paloukis and Theodore S. Papatheodorou

High Performance Information Systems Laboratory,
Computer Engineering and Informatics Dept.,
School of Engineering, University of Patras,
Building B, 26500 Patras-Rio, Greece
E-mail: kotsomit@hpclab.ceid.upatras.gr
E-mail: gpalouk@hpclab.ceid.upatras.gr
E-mail: tsp@hpclab.ceid.upatras.gr
*Corresponding author

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 representation produced 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.

Keywords: metadata; application profiles; semantic profiling; cultural heritage; ontologies; semantic web; inference; interoperability.

Reference to this paper should be made as follows: Koutsomitropoulos, D., Paloukis, G. and Papatheodorou, T.S. (2007) 'From metadata application profiles to semantic profiling: ontology refinement and profiling to strengthen inference-based queries on the semantic web', *Int. J. Metadata, Semantics and Ontologies*, Vol. 2, No. 4, pp.268–280.

Biographical notes: Dimitrios Koutsomitropoulos is a researcher at the High Performance Information Systems Laboratory (HPCLab), University of Patras. He is currently a PhD candidate at the Computer Engineering and Informatics Dept. from which he has received an MSc and a Computer and Informatics Engineering Diploma. His research interests include knowledge discovery, automated reasoning, ontological engineering, metadata integration, semantic interoperability and the semantic web.

George Paloukis is an MSc student at the Computer Engineering and Informatics Dept., University of Patras. He is member of the development and administration team of the institutional educational digital repository at the University of Patras. His research interests include metadata creation and use, metadata technologies and standards for educational resources, metadata application profiling and digital repositories.

Theodore S. Papatheodorou is a Professor at the Computer Engineering and Informatics Dept., University of Patras, since 1984 and now he is the chairman of this department. He has received a PhD in Computer Science in 1973 and an MSc in Mathematics from Purdue University in 1971 as well as a BSc in Mathematics from the University of Athens in 1968. He has authored hundreds of scientific publications in several areas of computer engineering and computer science. Recently, he has also co-authored papers published in the *Proceedings of the National Academy of Science of USA*, on protein folding.

1 Introduction

Ontologies play a key role in the Semantic Web idea. They serve as a means not only for conveying structural aspects and high-level data about information, but also for providing for its understanding and intelligent manipulation by a computer machine. Furthermore, ontologies on the Semantic Web are web-accessible and often distributed pieces of knowledge, a fact that at its own spawns a contemporary and challenging dimension in the way these ontologies are to be used, both from a technical as well as from a conceptual point of view.

Ontologies are often designed to depict a specific area of human knowledge, known as a *knowledge domain*. Most of the time such domain ontologies try to achieve a twofold goal: first, to be as thorough as possible, covering every potential aspect of the domain under consideration; and second, not to be extremely specific, in a sense that would not compromise the most general usefulness of the ontology. Besides this, according to Guarino (1998), an ontology may only approximate the conceptualisation of the domain knowledge.

The CIDOC Conceptual Reference Model (Crofts et al., 2003) is such an ontology that attempts to model the knowledge domain of cultural heritage. As of every human-conceivable domain, cultural heritage is very hard to be accurately modelled. In addition and owing to its nature, cultural heritage information are often hidden in libraries and museum archives, and when available online 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 of use as a common conceptual basis between cultural heritage applications.

On the other hand, the Semantic Web comes to offer a whole range of tempting possibilities, ranging from web knowledge management to semantic resource description to distributed knowledge discovery. Then, the elaborate representation of knowledge in an ontology combined with intelligent reasoning tools determine the extent to which one can deduct new and useful knowledge that is implied among the ontology lines.

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 (Bechhofer et al., 2004). This process does not merely amount to a simple syntax transformation. Rather, taking advantage of OWL most expressive but, simultaneously, decidable (Horrocks and Sattler, 2005) structures we also enrich and upgrade the model, thus further narrowing the conceptual approximation. The method and the lessons learned during both the syntactic and the semantic transformation are thoroughly documented.

However, one cannot go forever with this process; there is always the danger of rendering the model too specific, thus putting its applicability in risk. To avoid this, we incorporate the OWL-specific and powerful statements in different OWL 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; it also suggests a *semantically enhanced application-profiling* paradigm where the separating line between a standard and application-specific accuracy is thin and crucial. This kind of profiling takes the usual metadata-specific sense, where it is seen as an aggregation of disparate metadata elements (Duval et al., 2002), a step further: It does not deal so much with the horizontal extension of the ontology, but rather extends it in a semantic manner, as may be 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. As standard the language and the model may be, the process for doing this is not; thus we employ a methodology (Koutsomitropoulos et al., 2006) and implement a prototype web application, the Knowledge Discovery Interface (KDI) to be able to pose more expressive, reasoning-based intelligent queries to the CRM-profiled form.

The rest of this paper is organised as follows: First, in Section 2, we give an overview of the metadata application profiling idea and approaches. Then, in Section 3, we introduce the CIDOC Conceptual Reference Model, its structure and its semantics. Section 4 presents our process of transforming and profiling the CIDOC-CRM, pointing out our enhancements and discussing semantic profiling; following these are the inferences conducted on the CRM using the KDI and their results in Section 5. In Section 6, we discuss some previous work on the field of knowledge acquisition on the web, including efforts focusing on cultural heritage data. Finally, in Section 7, we summarise 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 organised in concrete metadata 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 a particular developer's needs were not satisfied by any existing schema or some elements she may find suitable were 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 Duval et al. (2002), application profiling is the assemblage of metadata elements selected from one or more metadata schemas and their combination in a compound schema. Application profiles provide the means to express principles of modularity and extensibility. The purpose of an application profile is to adapt or combine existing schemas into a package that is tailored to the functional requirements of a particular application, while retaining interoperability with the original base schemas.

There are a few ways of developing a particular metadata profile (Heery and Patel, 2000): most obvious is to include in the same schema selected elements from different standards suitable for the particular application. If new elements are to be defined, this has to be absolutely necessary and the new elements must refer to their own namespace. Another technique is to restrict value ranges of elements, e.g. provide a specific controlled vocabulary as filler to an element or mandate specific formats for values. Finally, a profile may refine existing elements defined in standards. This may involve, for example, the definition of sub-elements that intend to narrow the meaning of a definition or introduce some element qualifications.

Let us briefly examine an application-profiling example (Powell and Johnston, 2003):

```
<?xml version="1.0"?>
<record
xmlns="http://example.org/learningapp/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://example.org/learn
ingapp/
http://example.org/learningapp/schema.xsd"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ims="http://www.imsglobal.org/xsd/imsm
d_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, 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.

The Semantic Web and its ontologies give the chance of more accurate modelling of domain knowledge and thus upgrading metadata from a machine-readable to machine-comprehensible state. In fact, ontologies *are* 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 also by the *semantic enhancement* of the model, and that is exactly what we are trying to do in Section 4.

3 The CIDOC conceptual reference model

CIDOC-CRM may be considered as a domain ontology in the sense given by Guarino (1998). Thus, it covers only a focused area of interest and not the general knowledge. In addition, CRM has been designed to be extensible, flexible and implementation-independent. As a result, it can be easily harmonised with other upper-level ontologies or conceptual schemas to serve the modelling tasks of specific organisations, as well as the information integration needs from conceptually heterogeneous sources (Doerr et al., 2003).

In this section, we give a brief overview of the CRM structure, originally introduced in Crofts et al. (2003). Then, we discuss its machine-readable implementation(s) and the corresponding expressivities it provides for.

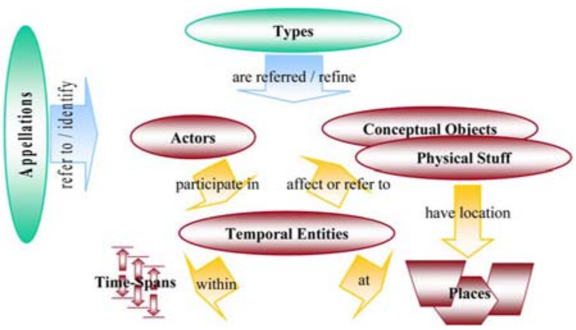
3.1 Conceptual structure

The CIDOC-CRM can be best described starting with the broad classes. These high-level classes are those that emerged as a result of the logical grouping of shared properties. These groups are concerned with fundamental notions such as identification, participation, location, purpose, motivation and use. Figure 1 presents an overview in which *Temporal Entities*, and hence events, occupy a central place.

All property paths to dates go through Temporal Entities, as do most of the property paths to places. Those place properties that bypass temporal entities should be understood as short cuts of temporal entities. Similarly, actors are only seen as relating material and immaterial things (Physical Stuff, Conceptual Objects) through Temporal Entities.

Any instance of a class may be identified by a number of appellations. These are the names, labels, titles or other means of identification used in the historical context. The ambiguous relation of items to their names is modelled as part of the historical process of knowledge acquisition. The notion of identification used here should not be confused with that of database identifiers in implementations of the model, which are not part of the ontology.

Figure 1 A qualitative meta-schema of CIDOC-CRM (see online version for colours)



Source: Doerr (2003)

All class instances can be refined (specialised) into more detailed categories through the use of types. Types frequently consist of a range of properties that refer in general to things of a certain kind, such as “a dress made for a wedding” in contrast to the “dress made for my wedding”.

CRM properties can be grouped by the following list of meta-properties:

- identification of real-world items by real-world names
- classification of real-world items
- part-decomposition and structural properties of Conceptual and Physical Objects, Periods, Actors, Places and Times
- participation of persistent items in temporal entities
- location of periods in space–time and physical objects in space
- influence of objects on activities and products and vice versa.

3.2 *Implementation and expressivity*

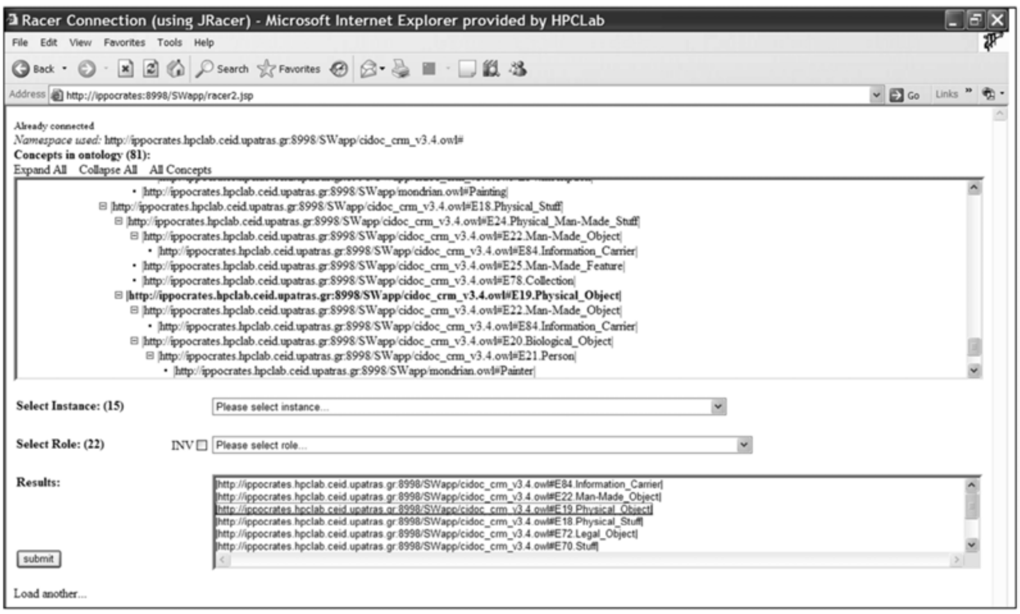
CIDOC-CRM is currently at version 3.4.10 (aka version 4). In our work, we used the initial 3.4 version, because this is the most up-to-date CRM’s version that maintains a machine-readable implementation. Later versions include small-scale updates regarding mostly insertion, deletion and renaming of concepts and roles in the model. Among its implementations, we chose RDF(s), as the semantically richest and closest to OWL available format.

As of January 2005, there exists an OWL transcription of the CRM’s RDF document (Balzer, 2006). However, this version adds only role-specific constructs (inversion, transitivity, etc.) which, semantically, do not exceed OWL Lite.

Version 3.4 includes about 84 concepts and 139 roles, not counting their inverses (that is, a total of 278 roles) (Figure 2). In terms of expressivity, the CRM employs structures enabled by RDF(s), which may be summarised as follows:

- concepts as well as roles are organised 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 data type properties (roles) as in OWL; rather, roles that are equivalent to data type properties have `rdf:Literal` as their range.

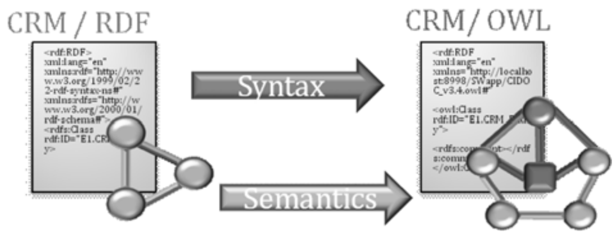
Figure 2 CIDOC-CRM taxonomy as shown by the KDI



4 Profiling the CIDOC-CRM

In the following, we discuss our process of transforming CIDOC-CRM in terms first of its syntactic transcription and then its semantic enhancement and profiling (Figure 3). 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 (Nardi and Brachman, 2007) and then by refining the model for the needs of a particular application.

Figure 3 The CIDOC-CRM transformation process



4.1 Transforming syntax

To transform the ontology to OWL syntax, we initially utilised the RACER system (Haarslev and Möller, 2003). RACER has the ability to load and process ontologies expressed in various formats, including RDF(s) and OWL. One can instruct RACER to load TBoxes expressed in RDF(s) by using the `rdfs-read-tbox-file` command. Once loaded, the TBox can then be exported to the appropriate format by using the `save-tbox` command along with the `:syntax` parameter.

Following these steps, we actually received a formal OWL document representing correctly the initial ontology. However, we discovered that RACER included some unnecessary and redundant statements, which, in many cases, were semantically overlapping. For example:

- For every role and concept, RACER included tags from the OILed namespace; in particular, RACER added the tags `oiled:creationDate` and `oiled:Creator`, which were neither required nor included in the initial document.
- For every concept defined as domain or range, RACER used the `owl:UnionOf` operand, thus expressing these restrictions as singleton concept unions (including only the concept in particular).
- The definition of role domains and ranges, even in OWL, comes from the RDF(s) namespace (`rdfs:domain, rdfs:range`). RACER, even though it maintains these statements, it duplicates them with equivalent expressions, which relate to the description logics (DL) style of expressing this kind of restrictions. These equivalent statements involve number and value restrictions and can be represented in OWL.

This process resulted in transforming the initial 60 KB file to a 478 KB OWL document. We, therefore, opted for the manual transcription of the RDF(s) document, during which common expressions between RDF(s) and OWL were preserved (e.g., `rdfs:subClassOf` and `rdf:resource`), while we replaced some namespace prefixes and updated the terminology used (e.g., `owl:Class` instead of `rdfs:Class` and `owl:ObjectProperty` or `owl:DataTypeProperty` instead of `rdf:Property`). In this manner, the CRM syntactical transformation phase was completed, resulting in a 63 KB document, named cidoc_crm_v3.4.owl.

4.2 Semantic intension and refinement

The second phase of CRM-upgrading process included 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. Table 1 summarises the expressivity gains of the semantic profiling technique on the CRM.

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 us examine these steps in detail:

4.2.1 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 conceptualisation 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.

To a certain extent, adding cardinality constraints to properties may be considered a profiling act, since the model clearly specifies that these quantifiers are provided only for semantic clarification. Nevertheless, by doing this we achieve the shift of intended meaning from inside text notes and annotations to a semantic commitment. Please also 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.

Clearly, the additions above actually refine the core model, either if this is intended in its specification or not. Profiling in this way, therefore, achieves to expand the *intensional knowledge* of the schema using constructs and means provided only in a Semantic Web infrastructure.

4.2.2 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 characterised the 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.

For example (see also Section 5):

```
<owl:Class rdf:ID="Painter">
  <rdfs:subClassOf
  rdf:resource="&crm;E21.Person"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty
      rdf:resource="&crm;P14B.performed"/>
      <owl:someValuesFrom
      rdf:resource="#Painting_Event"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Table 1 Expressivity gains of semantic profiling on the CIDOC-CRM specific to OWL

Construct used	*Example*	*Inferences supported*
Intension strengthening (*crm_core_profile.owl*)		

Cardinalities	`<owl:`**`cardinality`**` rdf:datatype =`
	`"&xsd;nonNegativeInteger">4`
	`</owl:cardinality>`
		Discover relation inconsistencies
		Infer instance equality (when cardinality = 1)

Inverse roles	**`<owl:ObjectProperty rdf:about="&crm;P7F.took_place_at">`**
	`<owl:inverseOf rdf:resource="&crm;P7B.witnessed"/>`
	`</owl:ObjectProperty>`
		Discover relations between instances

Symmetric roles	`<owl:`**`SymmetricProperty`**
	`rdf:about="&crm;P139F.has_alternative_form">`
	`</owl:SymmetricProperty>`
		Discover relations between instances

Application refinement (*crm_paint_profile.owl*)		

Concrete domains (data types)	`<owl:`**`DatatypeProperty`**` rdf:ID="hasURL">`
	`<rdfs:subPropertyOf rdf:resource= "&crm;P55F.has_current_location"/>`
	`<rdfs:range rdf:resource= "http://www.w3.org/2000/01/rdf-schema# Literal"/>`
	`</owl:DatatypeProperty>`
		Perform datatype reasoning
		Retrieve URI resources

Existential quantification	`<owl:Restriction>`
	`<owl:onProperty rdf:resource="&crm;P94F.has_created"/>`
	`<owl:`**`someValuesFrom`**` rdf:resource="#Painting"/>`
	`<rdfs:subClassOf rdf:resource="#Painting_Event"/>`
	`</owl:Restriction>`
		Define artificial classes and infer membership of instances based on their relations

Universal quantification	`<owl:Restriction>`
	`<owl:onProperty rdf:resource="&crm;P94F.has_created"/>`
	`<owl:`**`allValuesFrom`**` rdf:resource="#Painting"/>`
	`</owl:Restriction>`
		Infer instance relations based on their membership to the artificial class

Nominals	`<owl:Restriction>`
	`<owl:onProperty rdf:resource="&crm;P2F.has_type"/>`
	`<owl:`**`hasValue`**` rdf:resource="#painting_composition"/>`
	`</owl:Restriction>`
		Infer class membership
		Discover relations between instances
		Infer instance equality

This fragment defines a new class, namely ‘Painter’ and states that a ‘Painter’ is any ‘Person’ that has ‘performed’ at least one ‘Painting_Event’. Then, we can instantiate Mondrian as follows:

```
<crm:E21.Person rdf:ID="Mondrian">
 <crm:P14B.performed>
  <Painting_Event rdf:ID="Mondrian's
  Composition"/>
 </crm:P14B.performed>
</crm:E21.Person>
```

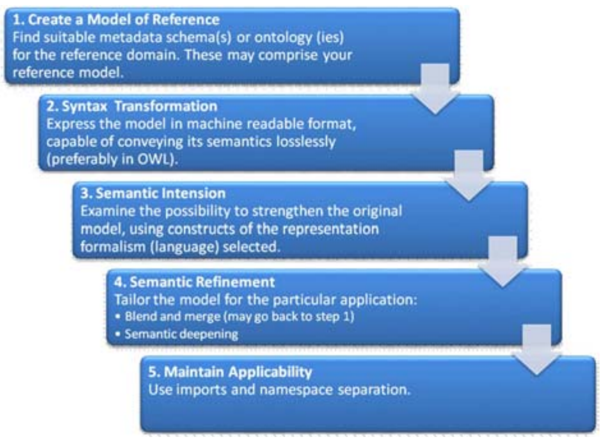
Given the ‘Painter’ class definition, it is straightforward for a Semantic Web reasoner to infer that ‘Mondrian’ is indeed a painter.

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.

4.3 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 easily be seen that it fits any other domain of interest. As shown in Figure 4, 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.

Figure 4 A process for developing semantic application profiles (see online version for colours)



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 conceptualisation 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 to narrow the semantics of the intended application.

One of the main concerns when developing an application profile is to ensure that 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.[1] In this way, we preserve the original model and we also show Semantic Web capabilities for ontology integration and distributed knowledge discovery.

Backward compatibility of the original model is also an important consideration that may be dealt with, using this approach. In its efforts to bring the Dublin Core Metadata Set to the Semantic Web reality, the Dublin Core Metadata Initiative (DCMI) is facing such a problem: By defining DC elements as RDF properties and assigning them restrictions, legacy metadata may appear invalid in the context of inferencing applications (Nilsson and Baker, 2007), as is the case for example with `dc:creator` and `dc:contributor`. To overcome this obstacle, the DCMI charter seems to follow a similar tactic (Nilsson et al., 2007): Semantically profile the DC model by defining domains and ranges and then maintain compatibility by using imports and separating namespaces. Of course, the DCMI group is more decisive in that it intends to *fully* refine the model and not just profile it for a particular application.

5 Results

In the following, we present the results from some experimental inference actions conducted on the CRM-profiled OWL form, so as to evaluate the ‘semantic performance’ of our profiling technique. To conduct these inferences, we use our KDI, which is briefly presented first.

The point is to investigate how much powerful reasoning (in terms of expressivity) is enabled with the use of the semantic profile, compared with the original model. To this end, it is worth noting that all of the inferences presented are *unique*, in the sense that they are possible only because of the involvement of the profile.

We conclude the results with a concrete usage case scenario, in which the user, through a series of inferences, is lead to the discovery and retrieval of an actual web resource.

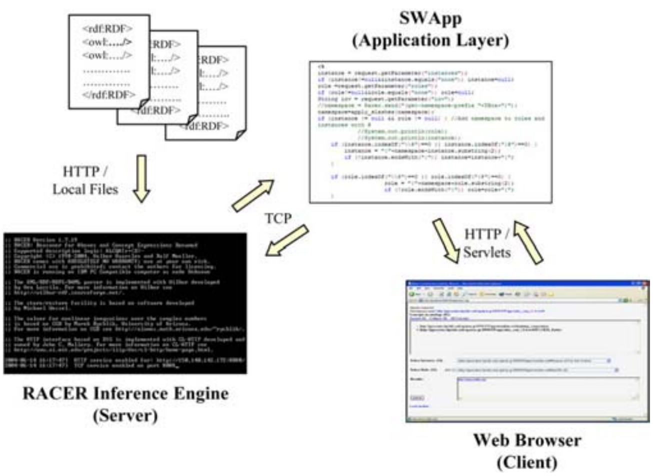
Inferences performed can be divided into two categories: *Positive inferences* where, based on the concept and role axioms as well as the ontology facts, we conclude new, not explicitly expressed facts, and *negative inferences* where, based on the ontology axioms and facts, we detect unsatisfiability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

For every example, we give the OWL fragment where the inference is based on, and we graphically explain the reasoning process in terms of DL formalism. To save space, instead of full namespaces we use the prefix &crm; for entities originating from the cidoc_crm_v3.4.owl document, &crm_p; for entities belonging to cidoc_paint_profile.owl and the default prefix '#' for entities coming from the mondrian.owl document (which includes the others). Relationships and assertions that hold in these results are also depicted graphically according to the following legend: a box denotes an instance, a circle stands for a concept, a headed arrow between instances i_1, i_2 a relation $R < i_1, i_2 >$, an arrow between concepts a subsumption relationship towards the direction of the arrow and an arrow between an instance and a concept denotes a membership ('isA') relationship.

5.1 *The Knowledge Discovery Interface*

The KDI is a web application, providing intelligent query submission services on web ontology documents. We use the word *Interface* to emphasise the fact that the user is offered a simple and intuitive way to compose and submit queries. In addition, the KDI interacts with RACER to conduct inferences. The interface design follows the traditional three-tier model, with an important variation: Where a database server would be typically used, we now use a knowledge base management system (Figure 5). Note that each of the three levels may be physically located on different computer systems.

Figure 5 The three tiers of the Knowledge Discovery Interface (see online version for colours)



The interface can load OWL documents that are available either on the local file system, or on the internet. A temporary copy of every document is stored locally on the application server and is then loaded by the knowledge base server (RACER). RACER creates and stores in memory an internal model for every ontology that it classifies. Classification takes place once for every ontology during its initial loading. Furthermore, other documents imported by the ontology may be loaded too.

The interface business logic was implemented using the Java programming language, as well as JSP, JavaBeans and Java Servlets technologies. Tomcat (version 5.0) was used as an application server. Business logic is mostly responsible for document loading, proper rendering of the ontological information to the user, composition and submission of queries and formulation of the results. Ontological data and reasoning results are fetched by interacting with RACER over the TCP/IP protocol. This interaction is greatly facilitated through the JRACER API. The latter has been modified in places, mainly in regard to the processing of web documents links and to the processing of synonym concepts.

The user interacts with the client front-end, where the appropriate JSP pages are rendered by the browser. Communication with the application layer is conducted over the HTTP protocol, using forms. At the same time, servlets are used for the administration of multiple user requests and for controlling simultaneous access. Furthermore, when a loaded ontology is not used any more, it is erased from memory to improve the utilisation of system resources. For a further description of the KDI, the reader is referred to Koutsomitropoulos et al. (2006).

5.2 *Positive inferences*

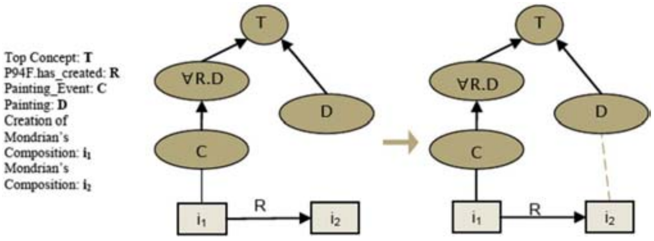
The following code is a fragment from mondrian.owl stating that a 'Painting_Event' is in fact a 'Creation_Event' that 'has_created' 'Painting' objects only:

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

The above fragment is graphically depicted in the left part of Figure 6. 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 6), correctly interpreting the value restriction on role R.

As simple as it may seem, this is indeed a very powerful inference. Without the value restriction on role 'has_created', the 'Mondrian's Composition' is just an instance of the world, i.e. it can be a book, a chair, a man or a time-period. It is just because of this restriction, apparent only in OWL DL, and thus made possible to express only after creating our semantic application profile, that 'Mondrian's Composition' is discovered to be a 'painting' and not anything else. In the next example, however, ‘Mondrian’s Composition’ is clearly stated to be a ‘painting’ from the beginning.

Figure 6 Inference example using value restriction (see online version for colours)

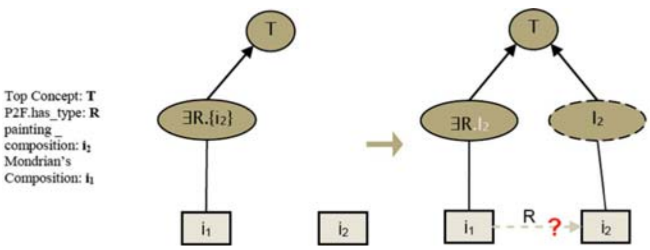


Let us now examine another example that involves the use of nominals (Horrocks and Patel-Schneider, 2003). The following fragment from Mondrian.owl states that a ‘Painting’ is a ‘Visual_Item’ that its ‘Type’ is ‘Painting_Composition’.

```
<owl:Class rdf:ID="Painting">
  <owl:subClassOf
  rdf:resource="&crm;E36.Visual_Item"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty
      rdf:resource="&crm;P2F.has_type"/>
      <owl:hasValue
      rdf:resource="#painting_composition"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<crm:E55.Type
rdf:ID="painting_composition"/>
<Painting rdf:ID="Mondrian's composition" />
```

The above fragment is graphically depicted in the left part of Figure 7.

Figure 7 Inference example using existential quantification and nominals (see online version for colours)



‘Mondrian’s Composition’ (i_1) is explicitly declared as a ‘Painting’ instance, which in turn is defined as a `hasValue` restriction on ‘has_type’ (R). ‘Painting_Composition’ (i_2) is declared as a ‘Type’ object. While the fact that ‘Mondrian’s Composition’ ‘has_type’ ‘Painting’ is straightforward, the KDI is unable to infer so and returns null when asked “what is the type of Mondrian’s composition?” (right part of Figure 7).

This example clearly demonstrates how difficult is for RACER as well as for every other current DL-based system to reason about nominals. Given the $\{i_2\}$ nominal, RACER creates a new synonym concept I_2 and makes i_2 an instance of I_2. It then actually replaces the `hasValue` restriction with an existential quantifier on concept I_2 and thus is unable to infer that $R(i_1, i_2)$ really holds.[2]

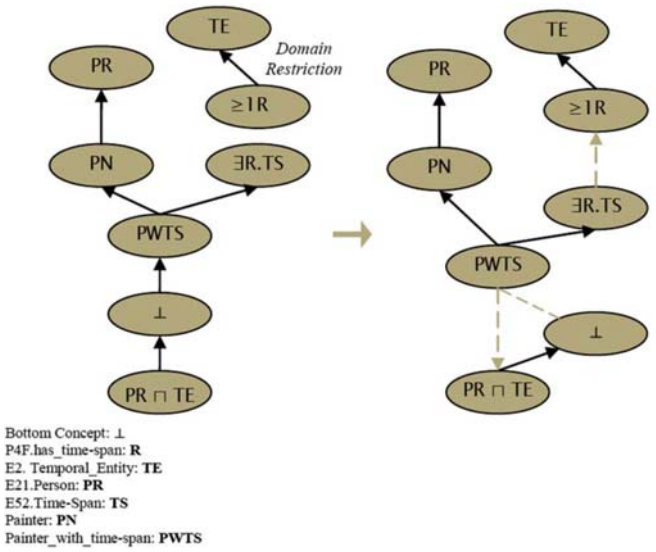
5.3 Negative inferences

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’.

```
<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 8. A ‘Painter with time-span’ is defined as a ‘Painter’ (known subclass of ‘Person’) that ‘has time-span’ some ‘Time-Span’ instances. However, individuals who ‘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.

Figure 8 Detecting unsatisfiable concepts (see online version for colours)



5.4 A usage scenario

In the following, we present a usage scenario, benefiting from our approach and involving the semantic application profile developed in Section 4 that deals with the world of painters and paintings. To do this, we link the ontologies to actual digital resources, using URI references inside the created OWL documents, thus imitating a virtual digital collection. These resources include digital surrogates of Mondrian's paintings.

We show how, through a series of inferences, the KDI can assist the user in discovering useful textual information as well as digital content (a digital image). In this scenario, taken for granted is of course the fact that the resources to be discovered are not fully described, i.e., there are missing parts in the discovery path to be filled in by inferences. It is evident that this situation, reproduced here, is representative of the semi-structured and incomplete (although proliferative) nature of web-distributed information and metadata.

As the detailed description of the user interaction with the KDI is out of the scope of this paper, we mainly stick to the inferences carried out and provide, for compactness, the DL counterpart of the ontology fragments involved in each of them.

Suppose first that our art-inclining user knows that a famous artistic movement existed, called 'De Stijl' but does not know much about it. First, she asks the KDI to find what 'isA' 'De Stijl', in other words, to perform instance checking on it (Table 2).

Because of the property domain restriction, 'De Stijl' is inferred to be a 'Group'; so who were its members? The KDI is asked to return the inverse relation on "P107B.is_current_or_former_member_of" using 'De Stijl' as the argument (Table 3).

Mondrian turns out to be a notable member of De Stijl, owing to the `inverseOf` declaration in the core profile. Our user knows Mondrian is an artist, but what has he dealt with? Is he a sculptor, a musician or a painter?

Table 2 *"What is De Stijl"?*

OWL document	*Snippet in DL syntax*
mondrian.owl	&crm;P107B.is_current_or_former_member_of (#Mondrian, #De Stijl)
cidoc_crm_v3.4.owl	⊤⊑∀&crm;P107B.is_current_or_former_member_of.&crm;E74.Group
Query submission and answer	
Instance	`|#De Stijl|`
Role	`isA`

Result	***`|&crm;E74.Group|`***
	`|&crm;E39.Actor|`
	`|&crm;E77.Persistent_Item|`
	`|&crm;E1.CRM_Entity|`

Table 3 *"Who are the members of De Stijl?"*

OWL document	*Snippet in DL syntax*
mondrian.owl	&crm;P107B.is_current_or_former_member_of (#Mondrian, #De Stijl)
cidoc_crm_v3.4.owl	&crm;P107B.is_current_or_former_member_of^{-} ≡ &crm;P107F.has_current_or_former_member
Query submission and answer	
Instance	`|#De Stijl|`
Role	`|&crm;P107F.has_current_or_former_member|`
Result	*`|#Mondrian|`*

As shown in Table 4, after a rather complex reasoning process, that also involves nominals, Mondrian is indeed found to be a 'Painter'. "Creation of composition1" has been 'carried_out_by' Mondrian and in turn 'has_created' composition1, which 'has_type' 'painting_composition'. Because of this and the nominal restriction on 'has_created' (similarly to Figure 7), composition1 is inferred to be a member of the 'Painting' class and so "Creation of composition1" is a 'Painting_Event'. Moreover, a 'Painter' is defined as a 'Person' who has 'performed' at least one 'Painting_Event' and 'carried_out_by' is the inverse of "performed", thus the conclusion follows.

Notice that nowhere in the ontology Mondrian is characterised as a 'Painter'. Without this reasoning path, he would just be an untyped resource/instance. In addition, all these details are withheld from the user.

Having found out his occupation, she now wants to know more about his works. She clicks on the result and the class 'Painter' is selected in the hierarchy. This has the effect that the roles available are reduced to only the most relevant ones, that is, the properties that have the selected class or its subsumers in their domain, displayed in a depth first search manner.

Table 4 *"What is Mondrian's occupation?"*

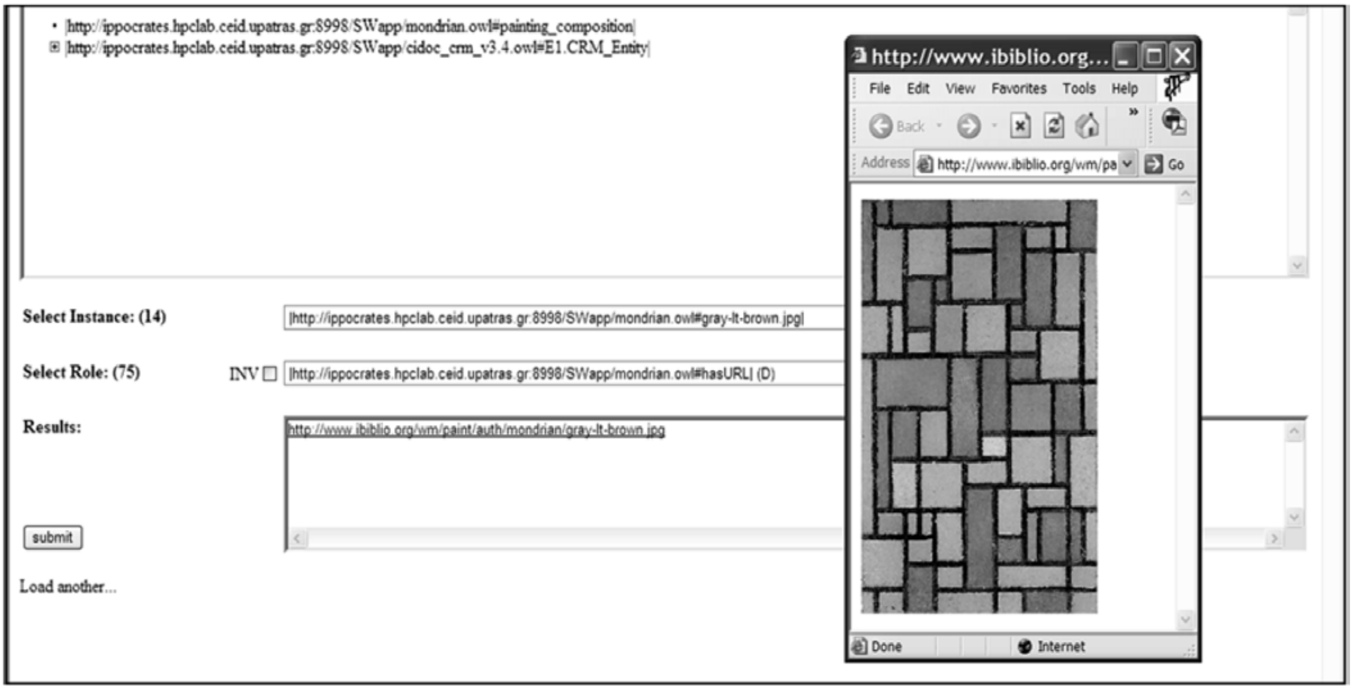
OWL document	*Snippet in DL syntax*
crm_core_profile.owl	&crm;P14F.carried_out_by$^-$ ≡ &crm;P14B.performed
crm_paint_profile.owl	#Painting≡ ∃&crm; P2F.has_type.{#painting_composition}
	∃&crm;P94F.has_created.#Painting⊑ #Painting_Event
	#Painter ≡ ∃&crm;P14B.performed.# Painting_Event
	&crm;P94F.has_created (#Creation of composition1, #composition1)
mondrian.owl	&crm;P2F.has_type (#composition1, &crm_p;painting_composition)
	&crm;P14F.carried_out_by (#Creation of composition1, #Mondrian)

Table 4 *"What is Mondrian's occupation?"* (continued)

Query submission and answer	
Instance	`|#Mondrian|`
Role	`|isA|`
Result	***`|&crm_p;Painter|`***
	...
	...

Based on the assertions shown in Table 4, the user finds out that composition1 is related to Mondrian and that it is a 'Painting', a defined subclass of 'Visual_Item'. So, where it can be seen? An inverse relation query on "shows_visual_item" reveals the file name "gray-lt-brown.jpg". Seeking for the file's location is straightforward using the 'hasURL' data type property that returns a clickable link, which, if followed, results in Figure 9.

Figure 9 Web resource retrieval using datatype properties



6 Related work

Even though the idea of the Semantic Web has only recently begun to standardise, the need for inference-based extraction and intelligent behaviour on the internet has long been a research goal. As expected, there have been some efforts in that direction. Such efforts include ontology description languages, inference engines and systems and implementations, based on them.

Simple HTML Ontology Extension (SHOE) (Heflin et al., 1998; Luke et al., 1996) was initially developed as an extension to HTML. It enables webpage authors to annotate their web documents with machine-readable knowledge. In that way, these documents can be more efficiently retrieved by knowledge-based search engines and then manipulated by agents. Although SHOE has a number of features, some of which are not present in other languages (e.g., n-ary relations), it lacks the expressiveness needed by the Semantic Web (for example, see Gomez-Perez and Corcho, 2002).

Knowing the constraints of knowledge discovery in a random environment like the internet, and taking into account the advantages of information retrieval, recent research has tried to combine these two approaches. OWLIR (Shah et al., 2002), for instance, is a system conducting retrieval of documents that are enriched with mark-up in RDF, DAML+OIL or OWL. A text-editing and extraction system is used to enrich the documents, based on an upper-level ontology. This extra information is processed by a rule-based inference system. Search is conducted using classical retrieval methods; however, the results are refined using the inference system outputs.

The TAP framework (Guha and McCool, 2003) seeks as well to improve the quality of search results by utilising the semantic relationships of web documents and entities. However, no inference takes place here. Instead, the RDF/OWL documents are treated as structured metadata sets. These sets can be represented as directed graphs, whose edges correspond to relations, and vertices correspond to existing internet resources. This representation is conducted based on the information of a local knowledge base.

Among the CRM applications, its use by the Artequakt system appears to be the most relevant to our work (Alani et al., 2003). Artequakt tries to alleviate the task of knowledge base maintenance by following an automated knowledge extraction approach. Artequakt applies natural language processing on web documents to extract information about artists and the artistic world and populate its knowledge base. Stored knowledge is then used for the automatic production of personalised biographies for artists. The CIDOC-CRM is used as the 'conceptual schema' for the information that needs to be extracted from the documents and stored in the knowledge base. Nevertheless, it should be noted that no inference – and thus knowledge discovery – takes place.

The Sculpture project (http://www.sculpteurweb.org) aims also at creating a semantic layer on top of a digital library of 3D cultural objects. Object properties and characteristics are organised with respect to the CIDOC-CRM ontology. Reasoning takes place within classifying agents that, when properly trained, are able to classify the objects in the ontology structure.

The Wine Agent system (Hsu and McGuinness, 2003) was developed as a demonstration of the knowledge discovery capabilities of the Semantic Web. This system uses a certain domain ontology written in DAML+OIL/OWL and performs inferences on it. The Wine Agent employs a first-order logic theorem prover (JTP).

Finally, a common motivation for our semantic profiling technique shares the approach of expressing application profiles with the OWL/XDD language (Ratanajaipan et al., 2006). By combining OWL constructs with rule-based expressions in XML syntax, it is claimed that application profiles with fine-grained semantic constraints may be represented. This method is then applied to define already modelled application domains, like for example the Dublin Core Library Application Profile (DC-Lib).

7 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 modelling domain metadata and be fruitful in that way.

At the same time, we have documented a procedure for knowledge acquisition which, having the CIDOC-CRM as a starting point, can be likewise applied in any other knowledge domain. To ensure feasibility and re-productivity, we developed and utilised suitable technical means for this, namely the KDI, as a proof-of-concept.

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 with specific needs.

A possible combination of semantic profiling with traditional metadata profiling practices like namespace inclusion and merging may be worth examining 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 specification (Cuenca Grau et al., 2006) and especially its concept of *punning* as a meta-modelling principle, based on which a name definition may have variable semantic interpretation depending on the ontological context.

Acknowledgements

The authors thank Detlev Balzer for his useful comments on the early parts of this work.

D. Koutsomitropoulos is partially supported by a grant from the 'Alexander S. Onassis' Public Benefit Foundation.

References

Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H. and Shadbolt, N.R. (2003) 'Automated ontology-based knowledge extraction from web documents', *IEEE Intelligent Systems*, Vol. 18, No. 1, pp.14–21.

Balzer, D. (2006) *Partial Definition of the CIDOC Conceptual Reference Model Version 3.4.9 in OWL*, Available at: http://cidoc.ics.forth.gr/official_release_cidoc.html

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P. and Stein, L. (2004) *OWL Web Ontology Language Reference*, W3C Recommendation, http://www.w3.org/TR/owl-ref/

Gomez-Perez, A. and Corcho, O. (2002) 'Ontology languages for the semantic web', *IEEE Intelligent Systems*, Vol. 17, No. 1, pp.54–60.

Crofts, N., Doerr, M. and Gill, T. (2003) 'The CIDOC conceptual reference model: a standard for communicating cultural contents', *Cultivate Interactive*, p.9, http://www.cultivate-int.org/issue9/chios/

Cuenca Grau, B., Horrocks, I., Parsia, B., Patel-Schneider, P. and Sattler, U. (2006) 'Next steps for OWL', *Proc. OWL Experiences and Directions Workshop (OWLED'06)*, Georgia, USA, http://owl-workshop.man.ac.uk/acceptedLong/submission_11.pdf

Doerr, M. (2003) 'The CIDOC conceptual reference model: an ontological approach to semantic interoperability of metadata', *AI Magazine*, Vol. 24, No. 3, pp.75–92.

Doerr, M., Hunter, J. and Lagoze, C. (2003) 'Towards a core ontology for information integration', *Journal of Digital Information*, Vol. 4, No. 1, p.169.

Duval, E., Hodgins, W., Sutton, S., Weibel, S.L. (2002) 'Metadata principles and practicalities', *D-Lib Magazine*, Vol. 8, No. 4, http://www.dlib.org/dlib/april02/weibel/04weibel.html

Guarino, N. (Ed.) (1998) 'Formal ontology and information systems', *Formal Ontology in Information Systems. Proc. FOIS'98*, IOS Press, Trento, Italy, pp.3–15.

Guha, R. and McCool, R. (2003) 'TAP: a semantic web platform', *Computer Networks*, Vol. 42, No. 5, pp.557–577.

Haarslev, V., Möller, R. (2003) 'Racer: a core inference engine for the semantic web', *Proc. 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*, Sanibel Island, Florida, USA, pp.27–36.

Heery, R. and Patel, M. (2000) 'Mixing and matching metadata schemas', *Ariadne*, p.25, http://www.ariadne.ac.uk/ issue25/app-profiles/intro.html

Heflin, J., Hendler, J. and Luke, S. (1998) 'Reading between the lines: using SHOE to discover implicit knowledge from the web', *AI and Information Integration: Papers from the 1998 Workshop*, AAAI Press, Madison, Wisconsin, USA, pp.51–57.

Horrocks, I. and Patel-Schneider, P.F. (2003) 'Reducing OWL entailment to description logic satisfiability', in Fensel, D., Sycara, K. and Mylopoulos, J. (Eds.): *Proc. 2003 International Semantic Web Conference (ISWC 2003)*, No. 2870 of LNCS, Sanibel Island, Florida, USA, pp.17–29.

Horrocks, I. and Sattler, U. (2005) 'A tableaux decision procedure for SHOIQ', *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland, pp.448–453.

Hsu, E. and McGuinness, D. (2003) 'Wine agent: semantic web testbed application', *Proc. 2003 Workshop on Description Logics*, Rome, Italy, URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-81/hsu.pdf

Koutsomitropoulos, D.A., Fragakis, M.F. and Papatheodorou, T.S. (2006) 'Discovering knowledge in web ontologies: a methodology and prototype implementation', *Proc. SEMANTICS 2006*, OCG, Vienna, Austria, pp.151–164.

Luke, S., Spector, L. and Rager, D. (1996) 'Ontology-based knowledge discovery on the world-wide web', in Franz, A. and Kitano, H. (Eds.): *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI96)*, AAAI Press, Portland, Oregon, USA, pp.96–102.

Nardi, D. and Brachman, R.J. (2007) 'An introduction to description logics', in Baader, F., Calvanese, D., McGuiness, D.L., Nardi, D. and Patel-Schneider, P.F. (Eds.): *The Description Logic Handbook*, Cambridge, pp.1–43.

Nilsson, M. and Baker, T. (2007) *Notes on DCMI Specifications for Dublin Core Metadata in RDF*, DCMI Note, http://dublincore.org/documents/dc-rdf-notes/

Nilsson, M., Powell, A., Johnston, P. and Naeve, A. (2007) *Expressing Dublin Core Metadata using the Resource Description Framework (RDF)*, DCMI Proposed Recommendation, http://dublincore.org/documents/dc-rdf/

Powell, A. and Johnston, P. (2003) *Guidelines for Implementing Dublin Core in XML*, DCMI Recommendation, http://dublincore.org/documents/dc-xml-guidelines/

Ratanajaipan, P., Nantajeewarawat, E., Wuwongse, V. (2006) 'OWL/XDD: a formal language for application profiles', *Proc. 2006 IEEE Conf. on Cybernetics and Intelligent Systems*, Bangkok, Thailand, pp.1–6.

Shah, U., Finin, T., Joshi, A., Cost, R.S. and Mayfield, J. (2002) 'Information retrieval on the semantic web', *Proc. 11th International Conference on Information and Knowledge Management*, McLean, Virginia, USA, pp.461–468.

Notes

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

[2]Clearly, a direct in-memory implementation, being interfaced by the appropriate OWL API, such as the ones provided by FaCT++ and Pellet would allow a successful answer to this kind of queries.