EXPRESSIVE REASONING ABOUT CULTURAL HERITAGE KNOWLEDGE USING WEB ONTOLOGIES

Dimitrios A. Koutsomitropoulos and Theodore S. Papatheodorou
High Performance Information Systems Laboratory, Computer Engineering and Informatics Dpt.
School of Engineering, University of Patras, 26500 Patras - Rio, Greece

Keywords: Ontologies, Reasoning, Cultural Heritage, Semantic Web.

Abstract: The cultural heritage knowledge domain is often characterized by complex semantic structures and a great lot of legacy information, possibly scattered on the Web that is not always properly structured. Thus, to achieve proper reasoning about this kind of knowledge one needs first a rather expressive model of representation that would also accommodate for its web distributed nature; and secondly a set of techniques that would allow for its intelligent and productive manipulation. The former can be served by the CIDOC-CRM which we first transform to the Semantic Web standard language, OWL and then augment with more expressive structures, possible only after this transformation. To show the latter we conduct a series of experimental inferences based on this CRM augmented form, using our Knowledge Discovery Interface. Our results clearly demonstrate the potential as well as the limitations of such an approach.

1 INTRODUCTION

The Semantic Web and its relating technologies gradually appear to proceed from a research and standardization experiment to a concrete and productive effort. As such, their application space has already started to span a wide range of domains, mostly because of the alluring capabilities promised: Web knowledge management, semantic resource description and distributed knowledge discovery are among the most important of them. Cultural heritage is such a domain, traditionally benefiting from the application of state of the art information technologies that assist and automate its documentation and information interchange needs. On the other hand, there is often skepticism around such efforts, grounded mostly on the fact that they do not always succeed in producing satisfactory and cost-effective results.

Recently, attention has been drawn to the CIDOC Conceptual Reference Model (CRM), currently under review by ISO. CIDOC-CRM (Crofts et al. 2003, Doerr 2003) is a reference ontology for the interchange and representation of cultural heritage information. It is mostly intended as a conceptual "template" for organizing, structuring and representing cultural information, rather than a concrete implementation of a knowledge schema. Nevertheless, it is also available in machine readable formats like XML and RDF.

Among the CRM applications, its use by the Artequakt system appears to be the most relevant to our work. Artequakt (Alani et al. 2003) tries to alleviate the task of knowledge base maintenance by following an automated knowledge extraction approach. Artequakt applies natural language processing on Web documents in order 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.

In this paper we examine the possibilities of applying Semantic Web techniques and ideas in order to enable reasoning on and discovery of cultural heritage information over distributed knowledge resources. Specifically, we show how to use the CRM, appropriately modified and extended for the Semantic Web environment, in order to perform useful inferences on cultural knowledge organized according to this model. First, we

A. Koutsomitropoulos D. and S. Papatheodorou T. (2007).
EXPRESSIVE REASONING ABOUT CULTURAL HERITAGE KNOWLEDGE USING WEB ONTOLOGIES.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 276-281
DOI: 10.5220/0001283502760281

transform and encode CRM to the Semantic Web standard language, OWL and present the lessons learned in this process. We then augment the model's expressivity by adding more expressive constructs made possible only after this transformation. We further complement CRM by adding some instances of CRM's concepts and roles, serving as a concrete modeling example. To be able to conduct our inferences, we have developed a prototype web based tool, the *Knowledge Discovery Interface* (KDI) that employs a reasoning module and aids the user to compose and submit intelligent queries to OWL documents, stored locally or on the Web. Using the KDI, we conduct a series of experimental inferences based on the CRM augmented form, which lead to the extraction of new, useful knowledge, not previously expressed in the ontology.

The rest of this paper is organized as follows: In section 2 we discuss our process of transforming and augmenting the CIDOC-CRM. Section 3 deals with the methodology that is actually followed to infer knowledge and introduces the KDI; then, section 4 shows the inferences conducted on the CRM and their results. Finally, section 5 summarizes the conclusions drawn from our approach.

2 UPGRADING CIDOC-CRM TO OWL

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 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.

Version 3.4 includes about 84 concepts και 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:

- 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.

Changes and extensions made to the RDF(S) CIDOC-CRM ontology, in order to upgrade to OWL, were performed in a two-phase procedure: First at syntactic and then at semantic level.

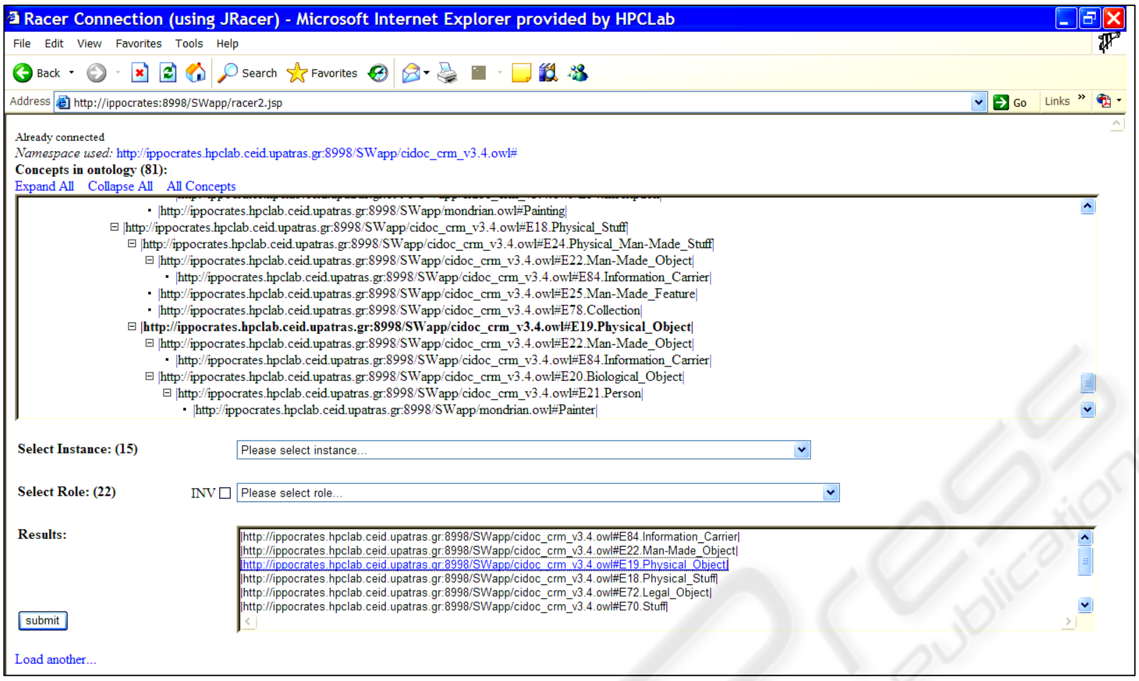
2.1 Transforming Syntax

In order to transform the ontology to OWL syntax, we initially utilized the RACER system (Haarslev & Möller 2003, Haarslev & Möller 2004). 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 not 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 DL-like 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 60KB file to a 478KB OWL document. We therefore opted for the manual transcription of the



http://ippocrates:8998/SWapp/racer2.jsp

Already connected

Namespace used: http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#

Concepts in ontology (81):

Expand All Collapse All All Concepts

- |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/mondrian.owl#Painting|
- ⊟ |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E18.Physical_Stuff|
  - ⊟ |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E24.Physical_Man-Made_Stuff|
    - ⊟ |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E22.Man-Made_Object|
      - |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E84.Information_Carrier|
    - |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E25.Man-Made_Feature|
    - |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E78.Collection|
  - ⊟ **|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E19.Physical_Object|**
    - ⊟ |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E22.Man-Made_Object|
      - |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E84.Information_Carrier|
    - ⊟ |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E20.Biological_Object|
      - ⊟ |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E21.Person|
        - |http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/mondrian.owl#Painter|

Select Instance: (15) Please select instance...

Select Role: (22) INV ☐ Please select role...

Results:

|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E84.Information_Carrier|
|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E22.Man-Made_Object|
|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E19.Physical_Object|
|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E18.Physical_Stuff|
|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E72.Legal_Object|
|http://ippocrates.hpclab.ceid.upatras.gr:8998/SWapp/cidoc_crm_v3.4.owl#E70.Stuff|

Figure 1: CIDOC-CRM taxonomy as shown by the KDI.

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 63KB document, named cidoc_crm_v3.4.owl.

2.2 Augmenting Semantics

The second phase of CRM upgrading process included its semantic augmentation with OWL-specific structures up to the OWL DL level, as well as its completion with some concrete instances. Although these extensions could have been integrated in the initial document, we chose to include them in a new file. The reason for this is to better show Semantic Web capabilities for ontology integration and distributed knowledge discovery.

More specifically, we created a document named mondrian.owl that includes CRM concept and role instances which model facts from the life and work of the Dutch painter Piet Mondrian. In this document we also included axiom and fact declarations that OWL allows to be expressed, as well as new roles and concepts making use of this expressiveness. In detail:

- We modeled minimum and maximum cardinality restrictions by using unqualified number restrictions (`owl:minCardinality`, `owl:maxCardinality`).
- We modeled inverse roles, using the `owl:inverseOf` operand.
- We included a symmetric role example, using the `rdf:type= "&owl;Symmetric"` statement.
- We constructed concepts based on existential and universal quantification, by using the `owl:hasValue`, `owl:someValuesFrom` and `owl:allValuesFrom` expressions, which ultimately enable more complex inferences.

The aforementioned documents were made available on the Internet through the Tomcat server. Inclusion of cidoc_crm_v3.4.owl axioms was possible simply by using the `<owl:imports>` directive in mondrian.owl. Therefore, loading mondrian.owl also loads all the axioms from cidoc_crm_v3.4.owl as well, as long as the latter is available on the Internet. In order to resolve potential ambiguities, different namespaces were defined for each document. In order to refer to statements from the imported ontology, the crm prefix is used, whereas for the new statements the default prefix (#) is used.

3 INFERENCE METHODOLOGY

Having expressed our ontology in OWL and created some typical instances, we should identify the means that would allow us to process this knowledge and deduct new facts out of it. In other words, reasoning support is explicitly needed to back the inference process. As OWL does not natively support or suggest a reasoning mechanism, we have to rely on an underlying logical formalism and a corresponding inference engine. In the following we discuss the use of *Description Logics* as the bottom line of our reasoning approach; then we introduce the KDI, the web service we have created to actually perform our inferences. This methodology is exhibited in more detail elsewhere (Koutsomitropoulos et al. 2006a, Koutsomitropoulos et al. 2006b).

3.1 Logical Formalism

Choosing an underlying logical formalism for performing reasoning is crucial, as it will greatly determine the expressiveness to be achieved. *Description Logics* (DLs) form a well defined subset of First Order Logic (FOL). OWL Lite and OWL DL are in fact very expressive description logics, using RDF syntax (Horrocks et al. 2003). Therefore, the semantics of OWL, as well as the decidability and complexity of basic inference problems in it, can be determined by existing research on DL. OWL Full is even more tightly connected to RDF, but its typical attributes are less comprehensible, and the basic inference problems are harder to compute (because OWL Full is undecidable). Inevitably, only the examination of the relation between OWL Lite/DL with DLs may lead to useful conclusions. On the other hand, even the limited versions of OWL differ from DLs, in certain points, including the use of namespaces and the ability to import other ontologies.

Horrocks & Patel-Schneider (2003) have shown how OWL DL can be reduced in polynomial time into *SHOIN*(D), while there exists an incomplete translation of *SHOIN*(D) to *SHIN*(D). This translation can be used to develop a partial, though powerful reasoning system for OWL DL. A similar procedure is followed for the reduction of OWL Lite to *SHIF*(D), which is completed in polynomial time as well. In that manner, inference engines like FaCT and RACER can be used to provide reasoning services for OWL Lite/DL.

On the other hand, neither the currently available Description Logic systems nor the algorithms they implement, support the full expressiveness of OWL DL. Even if such algorithms are implemented, their efficiency will be doubtful, since the corresponding problems are in NExp. Horrocks and Sattler (2005) have introduced a decision procedure for the *SHOIQ* Description Logic; this algorithm is claimed to exhibit controllable efficiency and is currently under implementation in two high-end inference engines.

Nevertheless, DLs seem to constitute the most appropriate available formalism for ontologies expressed in DAML+OIL or OWL. This fact also derives from the designing process of these languages. In fact, the largest decidable subset of OWL, OWL DL, was explicitly intended to show well studied computational characteristics and feature inference capabilities similar to those of DLs. Furthermore, existing DL inference engines seem to be powerful enough to carry out the inferences we need.

3.2 The Knowledge Discovery Interface

The KDI is a prototype web application, providing intelligent query submission services on Web ontology documents. We use the word *Interface* in order to emphasize 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. RACER was chosen because of its availability, its enhanced support for OWL DL as well as its ability to reason about the ABox.

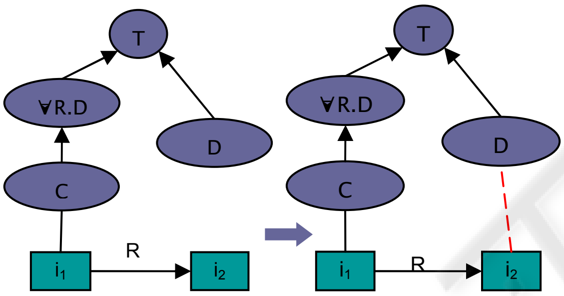
After connection to RACER has successfully been established, the ontology is loaded and its information is shown on the browser (see Figure 1). The user may navigate through the concept hierarchy, which is visualized in a tree form, and select any of the available classes. Upon selection, the page is reloaded, now containing in two drop down menus all of the instances of the selected class, as well as all of the roles whose domain is in this class. The user is able to select an instance and a role and then submit his query by pressing a button. Note that an option is available to invert the selected role, thus resulting in a different query.

We have identified such a declarative behavior to be of crucial importance for the Semantic Web knowledge discovery process; after all, the user should be able to pose queries even to unknown ontologies, encountered for the first time.

KDI helps the user compose a query by selecting a concept, an instance and a role in a user friendly manner. After the query is composed, it is decomposed into several lower level functions that are then submitted to RACER. This procedure is transparent to the user, withholding the details of the knowledge base actual querying and making the query composition process intuitive.

4 EXPERIMENTAL INFERENCES

In the following we present the results from a series of experimental inference actions conducted on the CRM augmented OWL form using our KDI. For every example we give the OWL fragment where the inference is based on, and we graphically depict the reasoning process in terms of the 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, as well as the default prefix “#” for entities coming from the mondrian.owl document (which includes the former).



Top Concept: **T**
P94F.has_created: **R**
Painting_Event: **C**
Painting: **D**
Creation of Mondrian’s Composition: **i_1**
Mondrian’s Composition: **i_2**

Figure 2: Inference Example using Value Restriction.

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>
```

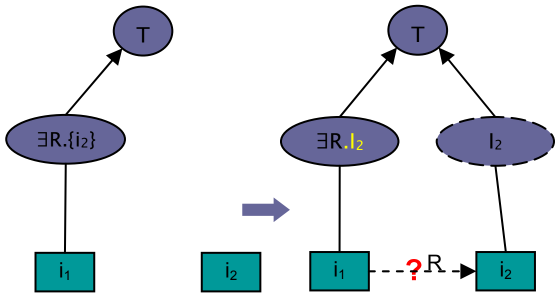
The above fragment is graphically depicted in the left part of Figure 2.

“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 3), correctly interpreting the value restriction on role R.

Let’s now examine another example that involves the use of nominals. 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 3.



Top Concept: **T**
P2F.has_type: **R**
Painting_Composition: **i_2**
Mondrian’s Composition: **i_1**

Figure 3: Inference Example using Existential Quantification and Nominals.

“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?"

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.

5 CONCLUSIONS

In this paper we have shown how to take advantage of the Semantic Web infrastructure in order to infer knowledge over the cultural heritage domain. As Semantic Web becomes a growing reality, domain modelers and specialists need to be prepared in order to adjust to this new environment and to rip the benefits of novel opportunities presented.

The CIDOC-CRM is identified as a key starting point for achieving cultural knowledge discovery. Based on the CRM, we have designated a process for representing cultural heritage information on the Semantic Web, by encoding the model in OWL and enriching it with more expressive semantic structures.

Furthermore we succeeded in conducting a series of inferences on web distributed cultural heritage information. The method we provide is grounded on a well-studied background and is based on decisions crucial for the quality, expressiveness and value of the inferences performed. In addition, the KDI demonstrates proper evidence of how this approach can be practically applied so as to be beneficial for a number of applications.

Our results seem to justify such an approach; at the same time they reveal that there are still limitations on the extent to which current state-of-the-art supports the full potential of the Semantic Web, especially in terms of its inferring capabilities. For example, the difficulty of current DL inferences engines to deal with nominals greatly hampers the expressiveness of our inferences.

Our results also suggest that augmenting the CRM with the OWL DL specific constructors leads to more powerful and semantically rich inferences. Thus, the incorporation of such "post-RDF" expressions in to the original model would probably lead to its better utilization by knowledge-intensive applications as well as to more accurate modelling of the domain.

ACKNOWLEDGEMENTS

Dimitrios A. 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*, 18(1): 14-21.

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

Doerr, M., 2003. The CIDOC conceptual reference model: an ontological approach to semantic interoperability of metadata. *AI Magazine*, 24(3): 75-92.

Haarslev V., and Möller R., 2003. Racer: A Core Inference Engine for the Semantic Web. In Proc. of the *2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*, pp. 27-36.

Haarslev V., and Möller R., 2004. RACER User's Guide and Reference Manual Version 1.7.19. http://www.sts.tu-harburg.de/~r.f.moeller/racer/ /racer-manual-1-7-19.pdf

Horrocks, I., and Patel-Schneider, P. F., 2003. Reducing OWL entailment to description logic satisfiability. In D. Fensel, K. Sycara, and J. Mylopoulos (eds.): P*roc. of the 2003 International Semantic Web Conference (ISWC 2003),* number 2870 of LNCS, pp. 17-29, Springer.

Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F., 2003. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7-26.

Horrocks, I., and Sattler, U., 2005. A tableaux decision procedure for SHOIQ. In Proc. of the *19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*.

Koutsomitropoulos, D. A., Meidanis, D. P., Kandili A. N., and Papatheodorou, T. S., 2006. OWL-Based Knowledge Discovery Using Description Logic Reasoners. 2006 Int. Conf. on Enterprise Information Systems (ICEIS 2006), SAIC track, pp.43-50.

Koutsomitropoulos, D. A., Fragakis, M. F., and Papatheodorou, T. S., 2006. A Methodology for Conducting Knowledge Discovery on the Semantic Web. In S. Sirmakessis (Ed.) *Adaptive and Personalized Semantic Web*, Studies In Computational Intelligence (14), pp. 95-105, Springer.