Ontology-Driven Processing and Management of Digital Rock Art Objects in IndianaMAS

Daniela Briola[1], Vincenzo Deufemia[2], Viviana Mascardi[1],
Luca Paolino[2], Nicoletta Bianchi[1]

[1] DIBRIS, University of Genoa, Italy
{daniela.briola,viviana.mascardi}@unige.it, nicoletta.bianchi@gmail.com
[2] DISTRA-MIT, University of Salerno, Italy,
{deufemia,lpaolino}@unisa.it

Abstract. This paper presents the Indiana Ontology for modeling the knowledge about Mount Bego's rock art and its exploitation in the IndianaMAS project. Although many projects use ontologies for semantic processing of cultural heritage digital objects, we are not aware of such ontologies in the rock art domain. Also, the Indiana Ontology is fully and seamlessly integrated with the IndianaMAS framework components, namely the intelligent software agents and the Digital Library used to classify and store multimedia and multilingual objects, thus making the IndianaMAS framework a good representative of an advanced and innovative information management system for the cultural heritage domain.

Keywords: Ontologies, Semantic Processing, Metadata, Digital Libraries, Rock Art.

1 Introduction and Motivation

Semantic-driven processing and management of information in the cultural heritage (CH) domain has been addressed by many scientists since the beginning of the new millennium and research projects, standards and ontologies related to this topic can be easily retrieved online[1]. However, when the domain of interest narrows to the rock art, results quickly narrow as well. Defining a general ontology for the rock art domain is difficult because the shape classification, meant as the association of a paradigmatic shape with a petroglyph or a group of petroglyphs, heavily depends on the place and time the rock art artifact was created, and no single ontology can fit all the possible "styles", even when referring to close places and ages. Things become even harder when we move from the shape classification to the petroglyph or scene interpretation.

---

[1] See for example [1, 2, 3, 4], or

`www.cs.vu.nl/STITCH/`,

`www.kulttuurisampo.fi/?lang=en`,

`europeana.eu/`,

`www.epoch-net.org`,

`http://www.cidoc-crm.org`, all accessed on September 2014

In this case, the ontology might include for example concepts like "female dancer, wizard, high goddess", as far as individual petroglyphs are concerned, and "ritual scene, pastoral scene, myth" when more petroglyphs composing a scene must be given a coherent meaning. Whereas stating that a petroglyph can be classified with a specific shape (for example, a "corniform") may be an almost easy task, stating that it represents the "high goddess" becomes more debatable, as the statement moves from the realm of shapes, which is objective, to that of intended meanings, more subjective. Different archaeologists may provide different interpretations to the same shape, and theories evolve quickly, requiring a constant update of the "interpretation" portion of the ontology.

Although these observations provide an explanation for the lack of ontologies in the rock art domain, processing and management of digital rock art artifacts would greatly benefit from an ontology-driven approach, supporting the automatic classification of multimedia and multilingual digital rock art objects. Separating the shape classification, which is stable and can be carried out by exploiting automatic techniques, and the interpretation, which is more dynamic and requires the formalization and application of rules derived from the archaeological theories to be made automatic, is mandatory to face the semantic processing of rock art images and, more in general, digital objects, feasible.

The *IndianaMAS* project [5] aims to digitally preserve the Mount Bego's rock art. In particular, the IndianaMAS framework exploits ontologies to provide a shared and human-readable representation of the domain [6], standard tools and technologies for digital libraries to manage and share digital objects [7], and intelligent software agents to automatically analyze, classify and give an interpretation to digital objects [8].

In this paper we discuss the design and implementation of the Indiana Ontology developed with archaeologists and domain experts, and its exploitation in the *IndianaMAS* project. The latter uses the Mount Bego archaeological site as a testbed: that area has been deeply studied during the last 100 years, and we based the Indiana Ontology on the work of Clarence Bicknell[2] and H. de Lumley[3], who led several teams for surveying and mapping the area starting from 1967 [9]. As for Bicknell, the University of Genoa owns many of his original manual drawings and reliefs, which have recently been published on the web[4] as one of the IndianaMAS outcomes and are going to be inserted into the IndianaMAS Digital Library (DL). We also possess a large portion of the reliefs made by de Lumley and his team, which were used to help the ontology design and to train our image classification algorithms, but which we cannot make public for copyright reasons.

This paper is organized in the following way: Section 2 presents the Indiana Ontology. Section 3 discusses the infrastructure that we designed and implemented in order to integrate the Indiana Ontology with the other core components of the IndianaMAS, namely the agents and the DL. Concluding remarks are presented in Section 4.

---

[2] Clarence Bicknell was a famous botanic and archaeologist that spent many years on the Mount Bego at the beginning of 1900, drawing reliefs of the petroglyphs.

[3] Henry de Lumley is the director of the Institute of Human Paleontology in Paris and Professor Emeritus at the Museum of Natural History in Paris.

[4] http://www.bicknell-legacy.it

2 The Indiana Ontology

The Indiana Ontology results from the collaboration between computer scientists and archaeologists who worked with H. de Lumley for many years, becoming some of the main experts worldwide on Mount Bego's rock art.

The ontology, implemented in OWL[5] using Protégé[6], is made up of three main parts: the first describes the classifications (related with petroglyph shapes) and interpretations (related with meaning, according to the most recent archaeological theories) of Mount Bego's most relevant individual and aggregated petroglyphs, the second models the types of digital objects (DOs in the sequel) managed by the IndianaMAS framework, and the third models the actions that intelligent software agents may perform. Another "branch", which we do not discuss here for space reasons, deals with geographical issues related with Mount Bego's sectors, zones, groups, and individual rocks.

Ontology for Petroglyphs Classification and Interpretation. The petroglyphs of Mount Bego can be classified into macro categories, which can be further refined into more detailed sub-categories, as described in [9]. In the Indiana Ontology, the tree structure induced by the is-a relationship has been limited to four levels of depth (considering the subclasses of Classification and Interpretation) inasmuch the algorithms for automatic classification of images representing petroglyphs and scenes, which we implemented in the project, reach that level of detail. For readability reasons, in Figure 1 we only show a portion of the ontology describing the possible classification and interpretation of Mount Bego carvings. The graphical representation of the full ontology can be found at http://www.bicknell-legacy.it/Ontology.aspx. Notice that, since in many cases petroglyphs may be composed by several symbols, in the following we will use the term *carving* to refer to one symbol of the physical petroglyph.

Archaeologists classify petroglyphs either by considering single carvings or by grouping them when they form a coherent and recurrent group. In Mount Bego, a single carving may represent an Anthropomorphic_Element (Figure 1(a)), a Corniform, a Weapon, an Agricultural_Tool or a Geometric_Figure (Figure 1(b)). A group of carvings may be a Group_of_corniforms, a Group_of_daggers, and so on, as shown in Figure 1(c). A deep and careful analysis has been conducted to classify Anthropomorphic_Element. It includes all the carvings similar to the human being. For example, Anthropomorphic_Corniform indicates those corniforms with some human traits, while Anthropomorphic_figure_made_of_corniforms represents figures with four appendages that could be identified as arms and legs. The subcategory Characters identifies carvings where human beings appear with some other carving, and it also contains subclasses to identify very specific figures that are rare but extremely relevant for understanding the cosmogonic and religious beliefs of the ancient people living in Mount Bego's region. Indeed, Characters subclasses are human figures with some particular traits that have been identified directly with an interpretation (based on [9]), such as the Wizard (Figure 1(a, d)).
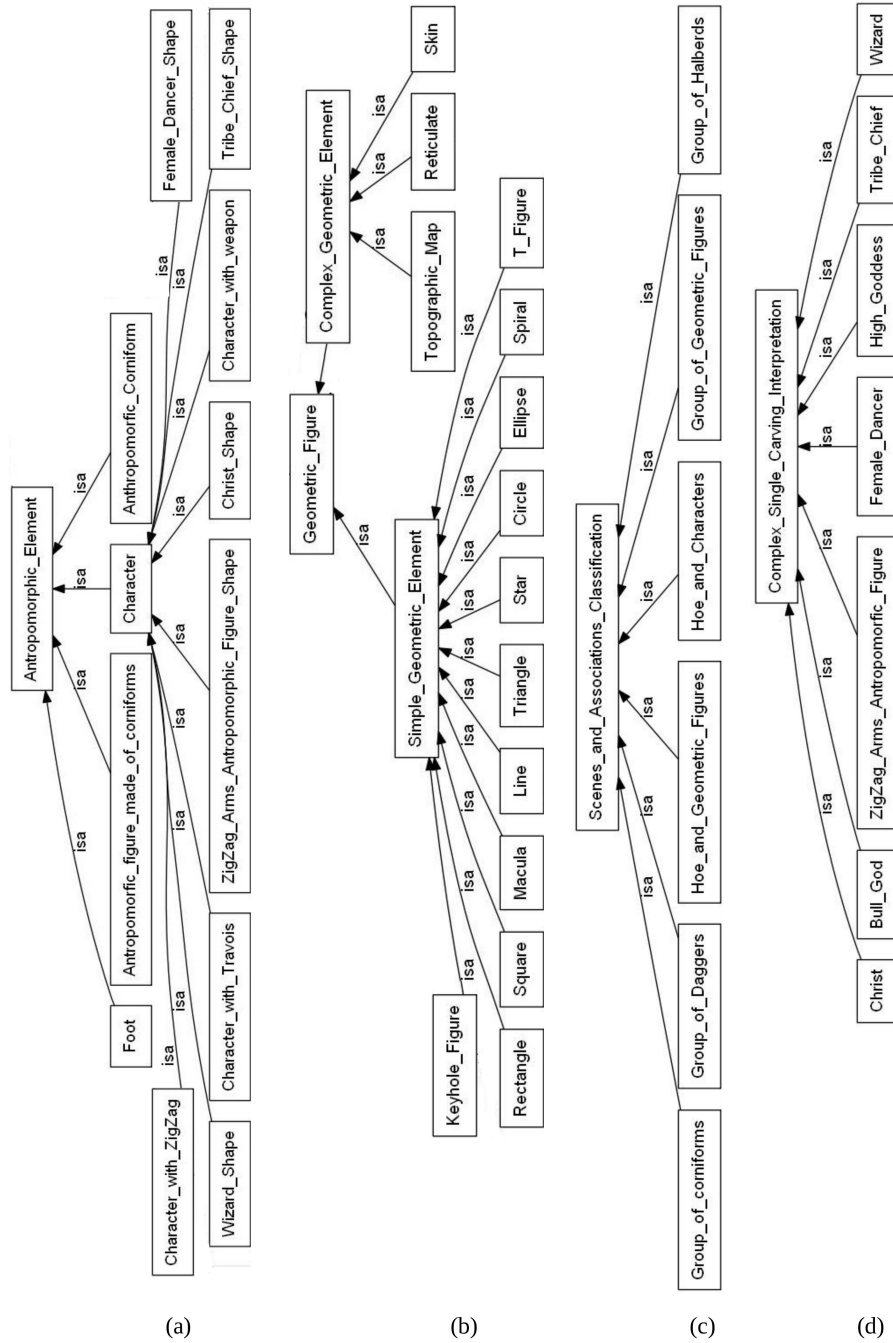

Fig. 1. A portion of the Indiana Ontology describing the classification of single carvings (a, b), the classification of groups of carvings (c) and the interpretation of complex single carvings (d). Sample images can be found in [9] and on http://www.bicknell-legacy.it.

Ontology for Representing Digital Objects. The IndianaMAS framework manages two types of DOs: texts and images. Manual sketches are a specific type of images. Texts may be multilingual and are classified according to the Indiana Ontology, whose concepts are expressed in English. For each DO type, different information has to be stored. Figure 2 reports the Indiana Ontology part modeling the DO types, their main properties and relationships. Note that *Object* properties are represented both as arrows connecting classes and as properties with the format: *Name of the property*, *Instance[*]*, *Range of the property*, while *Data* properties have the format: *Name of the property*, *Property type[*]*. The * after the term *Instance* or after the type (for example *String**) means that the property can be associated with more than one object in its range. In OWL, we were also able to add constraints over the properties, for example their cardinality, which are used by software agents to manage data correctly. These restrictions are not shown in the figure.
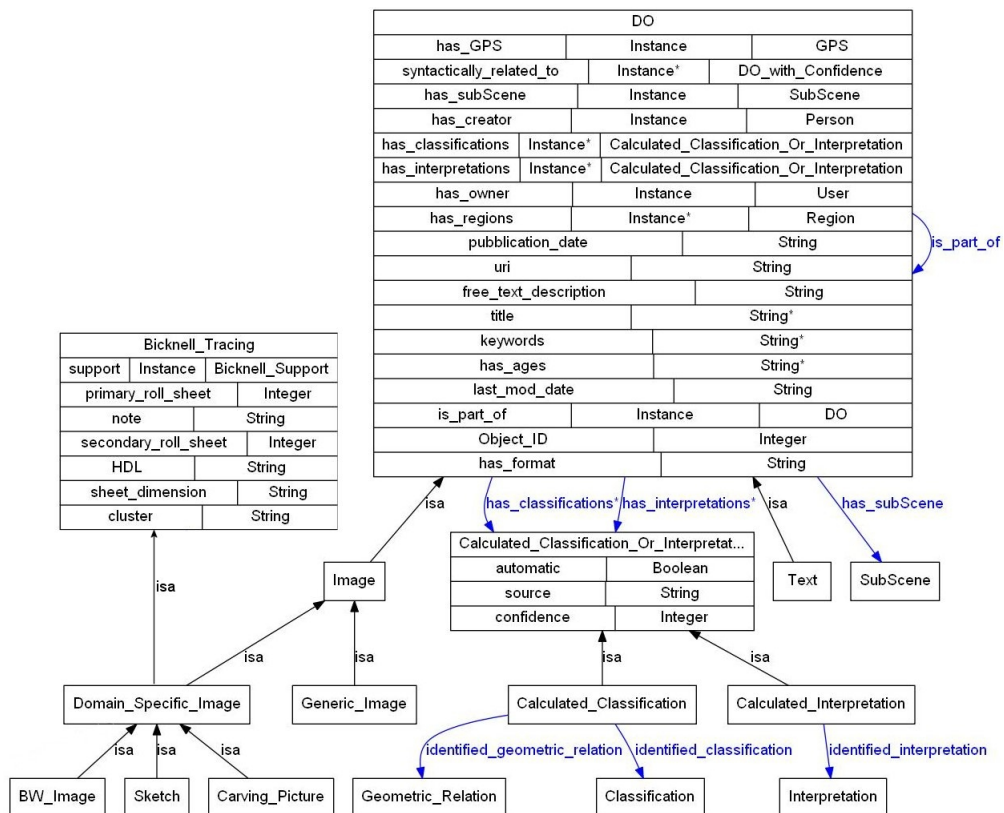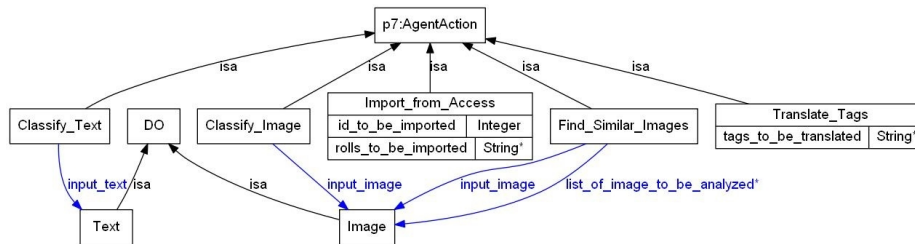

Fig. 2. The Indiana Ontology portion representing the main DOs and properties.

Ontology for Representing Agent Actions. The IndianaMAS framework is a multiagent system (MAS) developed in JADE (http://jade.tilab.com) and consists of intelligent agents that perform many different activities: end users interact with the

system through the IndianaGions DL, but the main operations over data (for example, the automatic classification) are performed in background by agents.

We use the Indiana Ontology to model the agents' actions and their arguments. JADE agents are able to exchange messages based on an ontology registered in the MAS: in this case, agents exchange Java objects representing an action to be performed (including its parameters if any) or the results of these actions, which are concepts of the ontology mapped in Java (as described in Section 3). Figure 3 shows the portion of the Indiana Ontology representing the agents' actions with their input parameters. The action results are not modeled in the Agent_Action class, and are usually a list of DOs or a list of Calculated_Classifications_or_Interpretations. For example, the action Find_Similar_Images takes in input a new Image (the one to be analyzed) and a list of Images (those existing in the system, to be compared with the one in input). In the agent action Translate_Tags, the only input is a list of Strings (the tags) to be translated.

Fig. 3. The Indiana Ontology portion representing agent actions. The name p7 is the name of an external ontology used by BeanGenerator, containing all the JADE classes needed for the ontology exportation.

3 Integrating the Indiana Ontology in IndianaMAS

Integration with the Digital Library. DOs managed by IndianaMAS are stored in a DL called *IndianaGions*, which has been created and managed with DSpace[7]. The latter equips each DO with a set of metadata composed of couples (*term*, *value*), as foreseen in any standard DL. In order to add the metadata describing the defined DOs, we mapped the OWL classes and properties to metadata and their values. In the DL community many standard metadata schemas exist, related to different areas. However, none of them is able to cover all the DOs' features defined in the previous section. As a consequence, we combined standard Dublin Core metadata (for example *title* or *owner*) with new domain specific metadata (for example *classification* and *interpretation*). Table 1 reports some of the resulting mapping from OWL to metadata.

Table 1. Mapping (part of) between OWL and Metadata. The OWL classes are in *italic*, and are reported to let the reader understand the OWL class the properties refer to.

| OWL Class or Property	Corresponding Metadata	Allowed Values
DO		
free_text_description	dc:description	String
has_interpretations	im:interpretation	Serialization of Calculated_Interpretation
has_classifications	im:classification	Serialization of Calculated_Classification
has_regions	im:regions	Regions
has_creator	dc:contributor.other	String
has_owner	dc.description.provenance	String
publication_date	dc:date.available	String
has_subparts	dc:relation.haspart	Serialization of SubScene
is_partof	dc:relation.ispartof	DO.Object_ID
semantically_related_with	im:semanticallyRelatedWith	Serialization of DO_with_Confidence
last_mod_date	dc:date.updated	Date
uri	dc:identifier.uri	String
	im:type	Text | Image
	im:imagetype	generic_image | sketch | bw_image| bicknell_tracing| carving_picture
Domain_Specific_Image. scene composition	im:sceneComposition	single_carving | many_carvings | scene_or_association
Bicknell_Tracing		
support	im:supportBicknell	Serialization of Bicknell_Support
primary_roll_sheet	im:primaryRollSheet	String
sheet_dimension	im:sheetDimension	String

Mapping OWL data properties to metadata was quite straightforward because they are transformed into a metadata with a string value, whereas the mapping of OWL object properties was much more complex. Metadata are flat data, while our domain objects are structured and complex objects, so we had to adopt a specific serialization of them. Moreover, since DLs are usually open to the community and can be dumped with standard protocols, a human readable format for metadata values is preferred, and in this case the standard object serialization is not usable. For example, a DO can have one or more classifications represented by the multiple property has_classifications. This property has been mapped into the metadata IndianaMAS:classification, which can be added many times, one for each classification, and its value, corresponding to the instance of OWL class Calculated_Classification, will have the format:

        carving: *subclass of Indiana ontology* Classification;

        confidence: *Number*; source: *String*;

        geometric_rel: *subclass of Indiana ontology* Geometric_Relation.

Similarly, the Interpretation property is serialized from class Calculated_Interpretation.

Integration with the JADE agents. The Protégé implementation of the Indiana Ontology allowed us to automatically export the OWL ontology into a set of Java interfaces and Classes with a format that JADE is able to directly import and manage inside the MAS. In particular, the Ontology BeanGenerator[8] plugin generates a Java representation of the ontology, i.e., for each OWL class, a Java interface and a Java class implementing it are created. In these classes, fields are automatically generated for each property of the OWL class with the correct type (that is, a simple type or an Object type), plus their setter and getter methods. Lastly, a new Java class is created to model the Java ontology, respecting the JADE formalism: in this class, all the operations to let JADE correctly register and manage the ontology are created. All these interfaces and classes are then packed in a jar file. As anticipated in Section 2, our JADE agents are already able to exchange messages based on the ontology: they use instances of the classes created with Ontology BeanGenerator, so that all the actions and messages contents are semantically related to the Indiana Ontology.

Interfacing with DSpace. Every DO in DSpace is represented with the *Item* class, whose structure can include one or more files and metadata. As a consequence, the Java *DO*s have to be converted into *Item*s and vice versa. The *DO2ItemConverter* in Figure 4 performs this task, allowing to convert the objects in both the directions. For storing DOs, the *DSpace REST Layer* takes in charge the items by translating them into an XML code specifically defined to be correctly parsed, and exploits the GET/POST/DELETE/PUT methods implemented in the REST wrapper for transferring the information from the MAS to DSpace. The latter is equipped with a REST module able to process the HTTP requests and to access the DL, as shown in Figure 4.
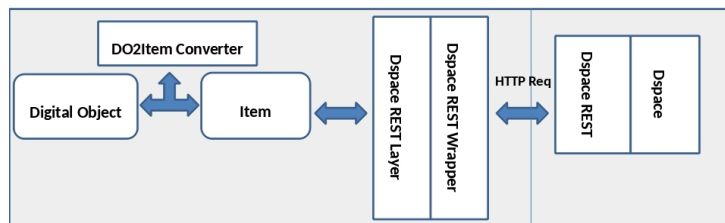
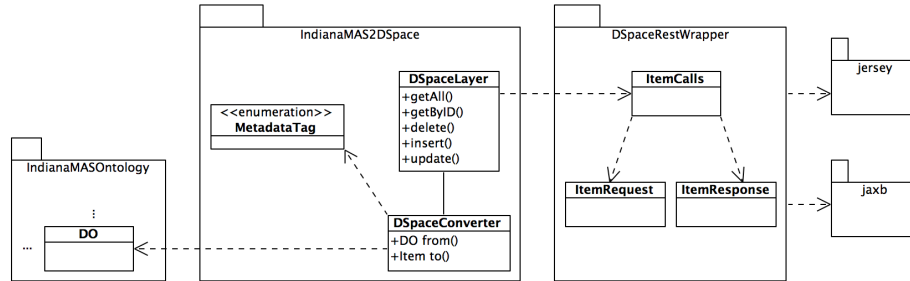**Fig. 4.** The architecture of the IndianaMAS module for storing and retrieving DOs within Dspace.

Figure 5 shows the class diagram of the implemented module. In order to store the items into DSpace and transmit them over the network, we transform the *Item* objects into XML format and wrap them into HTTP requests. To this aim, the *jersey*[9] and *jaxb*[10] libraries were used to implement the REST calls and to map the *ItemRequest* and *ItemResponse* classes into XML, respectively. Above them, the *ItemCalls* class implemented the connection to the DSpace REST module as well as the management of both the requests and responses.

---

[8] http://protegewiki.stanford.edu/wiki/OntologyBeanGenerator.
[9] https://jersey.java.net/
[10] https://jaxb.java.net/

Fig. 5. The class diagram showing the defined and used packages, with the relationships among the classes of the module.

To simplify the management of complex operations, we implemented a middleware layer named *DSpaceLayer*, which is useful to decouple functionalities from the specific implementation in a sort of DAO pattern. This class, together with the one managing the conversion between *Do*s and *Item*s, is implemented within the *IndianaMAS2DSpace* library. The conversion class, named *DSpaceConverter*, contains two main methods: *from* and *to*. The former transforms the *ItemResponse* objects obtained from the *get* methods into *DO* objects. The latter converts *DO*s into *ItemRequest* objects, namely extracts the fields values of the source object and insert them into a metadata structure (an array of elements composed by name and value fields), thanks to *MetadataTag* enumeration.

4 Conclusions

The adoption of an ontology as central point of the IndianaMAS system turned out to be a very good architectural solution to mitigate the complexity of the system itself. Thanks to the expressive power of OWL, we were able to model the Mount Bego domain (carvings classification and interpretation) with as many details as the archaeologists needed, and at the same time we were also able to extensively model the DOs and their properties, all in the same formal representation language. Then, thanks to the software applications and plugins available to the developers community and mentioned in Section 3, we could transform the Indiana Ontology into a Java jar, amenable for software employment, and into graphical representations too, amenable for human readability. Furthermore, the OWL language can be directly used in any JADE MAS, and the portion of ontology related with agent actions was a perfect formal representation usable by both humans and agents. With this ontology-centric solution, we have a unique point in the system where all the formal information of domain could be stored, so that it becomes easier to maintain the alignment between the many independent components of the MAS.

From the user point of view, having an ontology that guides him in the classification and query phases is a step toward the semantic usage of data, helping him in analyzing DOs, managing the complexity of the domain, and inferring new information.

Of course, the IndianaMAS ontology is not the only one in the CH domain. As a widely used representative of such ontologies, we may mention the CIDOC Conceptual Reference Model (CRM) [10] which provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation. Whereas the CIDOC CRM is an upper level ontology, covering knowledge which crosses all the museums, the IndianaMAS ontology is a very specific one, modeling information of a particular type of artifact, belonging to very specific regions both in space and in time. The IndianaMAS Ontology might be an extension of CIDOC, which already models concepts like Physical Man-Made Thing, suitable to represent rock art artifacts. However, since in the IndianaMAS project we do not need all the power of the CIDOC CRM, we preferred to model our domain as a standalone ontology.

Despite the existence of many ontologies in the CH domain, to the best of our knowledge the exploitation of an ontology fully integrated with a DL, and representing not only the domain of interest but also the actions available to the software components in the system, is an innovative and novel approach which distinguishes our project from similar ones.

References

1. Baruzzo, A., Casoto, P., Challapalli, P., Dattolo, A., Pudota, N., Tasso, C.: Toward semantic digital libraries: Exploiting Web 2.0 and semantic services in cultural heritage. Journal of Digital Information 10(6) (2009)
2. Hyvönen, E.: Publishing and using cultural heritage linked data on the semantic web. Synthesis Lectures on the Semantic Web. Morgan & Claypool Pub. (2012)
3. Pattuelli, M.C.: Modeling a domain ontology for cultural heritage resources: A user-centered approach. JASIST 62(2) (2011) 314-342
4. Stalmann, K., Wegener, D., Doerr, M., Hill, H.J., Friesen, N.: Semantic-based retrieval of cultural heritage multimedia objects. Int. J. Semantic Computing 6(3) (2012) 315-328
5. Mascardi, V., Briola, D., Locoro, A., Grignani, D., Deufemia, V., Paolino, L., Bianchi, N., de Lumley, H., Malafronte, D., Ricciarelli, A.: A holonic multi-agent system for sketch, image and text interpretation in the rock art domain. IJICIC 10(1) (2014) 81-100
6. Gruber, T.R.: A translation approach to portable ontology specications. Knowledge Acquisition 5 (1993) 199-220
7. Candela, L., Athanasopoulos, G., Castelli, D., El Raheb, K., Innocenti, P., Ioannidis, Y., Katifori, A., Nika, A., Vullo, G., Ross, S.: The digital library reference model. Technical Report Deliverable D3.2b, Project Number 231551. (2011)
8. Jennings, N.R., Sycara, K.P., Wooldridge, M.: A roadmap of agent research and development. AAMAS Journal 1(1) (1998) 7-38
9. de Lumley, H., Echassoux, A.: The rock carvings of the chalcolithic and ancient bronze age from the Mont Bego area. The cosmogonic myths of the early metallurgic settlers in the southern alps. L'Anthropologie 113(5) (2009) 969-1004
10. Doerr, M.: The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. AI Magazine 24(3) (2003) 75-92.