The Semantic Web and Cultural Heritage: 
Ontologies and Technologies 
Help in Accessing Museum Information

Oreste Signore 
C.N.R.-ISTI 
Area della Ricerca di Pisa - Via G. Moruzzi, 1 - 56124 Pisa  
Email: Oreste.Signore@isti.cnr.it  
personal home page: http://www.weblab.isti.cnr.it/people/oreste/

Abstract
A virtual museum should support rich semantic associations. In the past much effort has been devoted to match scholar needs, implementing appropriate aids. However, efforts toward a unified schema have all failed. Integration is often attempted at metadata level, but a more useful effort is to attempt to create a "core ontology" which incorporates basic entities and relationships common across the diverse metadata vocabularies. A core ontology (like CIDOC Conceptual Reference Model) is a formal model for automated tools that can be reasoned over, enabling the process of enriching knowledge. XML data structuring is semantically poor, while Semantic Web stack higher levels technologies (RDF, OWL, etc.) can supply the appropriate technical environment to represent, export and share the knowledge needed to implement intelligent retrieval and reason upon data. Semantic Web technologies allow fully decentralized semantic markup of content, and intelligent software agents can then use knowledge expressed by the markup. Web searching and browsing can appropriately link information, understanding the user's mental model and implementing suitable navigation mechanisms.

1 Introduction
A virtual museum should give the opportunity to go far further than the mere visualization of the museum content. The user will build "virtual exhibitions" putting together object belonging to other museums, but related, as can be the case of works of art fragmented or physically accessible in their original place, or combining information and items stored in different rooms. As cultural heritage is inherently rich in semantic associations, a virtual museum must support an interdisciplinary approach, through implementation of such semantic associations, allowing the user to understand the culture that is behind the objects and contextualize them.
Web technologies are a key for implementing and offering virtual museums to a wide audience. The World Wide Web Consortium\(^1\) (W3C) issued technical specifications, called W3C Recommendations\(^2\), covering the full range of the user needs, which fit well in the W3C long-term goals for creating one World Wide Web, namely: Web for Everyone, Web on Everything, Knowledge Base, Trust and Confidence.

Interoperability offers well known advantages, and is supported by conformance with a consistent framework/technology. However, it is not a mere technological issue. In particular, we should overcome differences in culture, language, education, ability, material resources, and physical limitations of users on all continents. This goal immediately leads to the issue of accessibility for impaired persons, but also, as the Web should really be a World Wide Web, to the issue of cultural barriers. Hence we have to consider differences in cultures and perceptions of concepts, going behind technological interoperability, to achieve semantic interoperability as well.

The rest of paper is organized as follows: section 2 discusses some issues concerning the cultural heritage applications. Section 3 will briefly explain matters related with semantic interoperability and ontologies. After a brief description of the Semantic Web technologies (section 4), we will discuss the advantages of an ontological approach (section 5). Finally section 6 describes possible scenarios.

2 Cultural Heritage Applications: some issues

2.1 Content and Users issues

A main issue regarding the content of cultural heritage web sites is the inherent associativity of information pertaining this field. Information must be put in the appropriate context, would it be geographic, cultural, or temporal. Represented subjects also can trigger some associations, driven by iconographic classification affinity. Cultural heritage must be seen as as multidisciplinary environment. While the scholar is normally familiar with geographic, historical or religious context end evolution, the generic user will need some support to better understand and appreciate the actual value of the information (s)he is gathering.

There is a large variety of potential users accessing cultural heritage information or museum content, ranging from professionals to practitioners or researchers or tourists (with different skills and competence levels). Any user can be interested in some context, best supported by an appropriate interaction metaphor (spatial, temporal, cultural, etc.), and have, in different circumstances, different search needs, varying from searching very specific documents (high precision) to looking for loose or marginal references (high recall). It becomes evident that there is need for adaptive and intelligent systems. An

\(^1\)http://www.w3.org/
\(^2\)See http://www.w3.org/TR/ for a complete list.
usual approach is to define several "user profiles" which however are static and often misused if not unused at all. We think that user profiles should be semantically described specifying which are the actual interests in terms of associations to follow, context of interest, etc.

2.2 Conceptual issues

There are several conceptual issues emerging when implementing applications in the cultural heritage area.

A first one, dating since long time, is about data structuring. For decades scholars have attempted to find an appropriate schema to describe works of art, aiming at a common schema to exchange and access information. Many efforts have been concentrated mainly at the purely syntactical level, sometimes giving less importance to the conceptual level [Signore1991].

A companion of the effort toward data structuring has been language normalization. This activity lead to a huge number of dictionaries, authority files and thesauri, which however result to be incompatible.

Finally, the spatio-temporal context is a major issue, as we have to store time-dependent information to be able, for example, to display chronologically consistent maps, or answer queries like "painters born in the Granducato di Toscana". In the cultural heritage context dates are often indetermined, and spatial context is time dependent (political and ecclesiastical jurisdictions vary in time). It is usual to have dates like: beginning 1800 or mid 19th century, or circa 1590, and so on. Note that we have to manage two different types of dates, simple dates and interval dates. The first one occurs when an event took place in some "point of time", which however can fall in an approximation interval. The second case is when the event has a duration, having therefore a time interval with a minimum and a maximum, both (possibly) known with some uncertainty. Hence the need for a temporal algebra which could support operators like: isBefore, in, partiallyOverlaps between dates or time intervals. In passing, we note that chronological ordering of fuzzy dates is a non trivial semantic issue. A solution to this issue has been proposed in [Signore1990] and [Signore1997].

2.3 Implementation issues

Among the major implementation issues we recall: links, dictionaries and authority files, classification systems and thesauri, interaction metaphors, which we will briefly discuss in the following.

2.3.1 Links

We can distinguish links in two main categories: structural and associative. This distinction has been stressed since the beginning of the hypertext explosion [DeRose1989]. Structural links are inherent to the specific item, and can be easily implemented in any environment as extensional links, explicitly stored
in the document. However "all the links are equal, but some links are more equal than others". In fact, associative links are the real value for the users, as they allow to contextualize the objects. These are the intensional links, driven by different association mechanisms. As noted before, we have many of them, and therefore it would be almost impossible, and certainly disturbing because of the consequent cognitive overhead, to explicitly store them in the document itself. The most obvious way of implementing these extensional links is via a "conceptual level". In this option, the user can navigate in the "data level" and follow intensional links up to the "conceptual level", where (s)he can navigate to find related concepts, and then go down back to the data level [Signore1995], [Signore1996].

2.3.2 Dictionaries and Authority files

Terminological dictionaries have been a long lasting effort, and are needed to support more "precise" queries, so reducing the silence effect in retrieval (low recall). The main difficulty is about their definition and agreement upon terminology, while their implementation is a trivial task.

Authority files are somehow more complex, as they are not just a list of terms, but a set of related info to rely upon. The correct search term can be found browsing and selecting a set of records. An example is the Italian Authors Authority file, where all the authors are listed together some additional information, like alternative names, dates of activity, roles played (painter, sculptor, etc.), school. Browsing this authority file, the user can get info about authors, select one of them, and also searching by school, for example3.

In passing, it is worthwhile to stress how the intrinsic, multicultural nature of cultural heritage leads to a multilingual issue. In fact, while no one, in any other application environment, would translate the last name of a person, different identifiers for the same author exist in current practice (e.g. the Italian artist 'Michelangelo' is known as 'Michel-Ange' in France).

2.3.3 Classification systems and Thesauri

A powerful IT supported example of a hierarchical classification system is Iconclass. Iconclass4 is a subject-specific classification system; it is a hierarchically ordered collection of definitions of objects, persons, events and abstract ideas that can be the subject of an image. Art historians, researchers and curators use it to describe, classify and examine the subject of images represented in various media such as paintings, drawings and photographs. The three main component of Iconclass are: the Classification System, the Alphabetical Index and the Bibliography. The Classification System is made by 28,000 hierarchically ordered definitions divided into ten main divisions. Each definition consists of an alphanumeric classification code (notation) and the description of the iconographic subject (textual correlate). The Alphabetical Index is made by 14,000

---

3See http://www.weblab.isti.cnr.it/talks/2006/ITVM2006/#[12]
4http://www.iconclass.nl/
keywords used for locating the notation and its textual correlate needed to de-
scribe and/or index an image. The Bibliography contains 40,000 references to
books and articles of iconographical interest.

According to [ISO2788] a thesaurus is a controlled list of terms linked to-
gether by semantic, hierarchical, associative or equivalence relationships. The
general principles in ISO 2788 are considered language- and culture-independent.
As a result, [ISO5964], refers to ISO 2788 and uses it as a point of departure
for dealing with the specific requirements that emerge when a single thesaurus
attempts to express 'conceptual equivalences' among terms selected from more
than one natural language.

A thesaurus can be represented as a graph, where nodes are the thesaurus terms,
and edges model semantic associations. A thesaurus can be seen as a simple
and straightforward implementation of the "concept space".5

An example of a controlled vocabulary is the Art Architecture Thesaurus
(AAT)6, one of the Getty Research Institute's Vocabulary Databases. It is
a hierarchical vocabulary of 123,000 terms for 31,000 concepts describing art,
architecture, and related fields.

2.3.4 Interaction metaphors

Users will interact and combine information using different metaphors. We can
reduce them to basically: time, space (geographic or topographic maps), classi-
fication, searching by access points, and their combination. All these metaphors
can be implemented creating a "conceptual level" which will support the inten-
sional links they are based upon. [Papaldo1989], a pilot project from which
the TGN7 was born, shows a typical example of (historical) map interaction
metaphor8, where an encoded representation of dates is used.

3 Information integration: role of ontologies

Data integration is concerned with some general and often neglected consider-
ations, mainly related to cross culture and internationalization issues. Names,
dates, colors etc., all can have a different meaning in a multicultural distributed
environment. For example, dates are based on different calendars in different
cultures (western, Islamic, Jewish), and, even in the same culture, like the west-
ern one, USA and European formats differ. Internationalization is also an issue,
as different alphabets or writing directions (left to right or right to left) can
be needed. Finally, we can’t ignore that in presenting information an implicit
knowledge is often assumed. At the inaugural Museum and Web conference in
1997, [Fink1997] presented a possible scenario for preserving our cultural her-
itage and enhance the world’s access to it. The envisaged scenario for 2005 was

5See http://www.weblab.isti.cnr.it/talks/2006/ITVM2006/#[18] for an example of
navigation driven by iconographic association.

6http://www.getty.edu/research/conducting_research/vocabularies/aat/

7http://www.getty.edu/research/conducting_research/vocabularies/tgn/

8See http://www.weblab.isti.cnr.it/talks/2006/ITVM2006/#[17]
that successful models for integrating our cultural heritage would exist, and any user would be "able to search the online universe seamlessly as if the images and text about culture were available in one vast library of information". The paper also pointed as technology is not the chief barrier to this vision, rather, the main obstacle is the need for cultural organizations to become willing to collaborate and form new partnerships. "Working alone, we can produce a lot of impoverished weed patches that, given the competition from the business and entertainment sectors, no one will want to visit. Working together, we can create a magnificent garden with something for everyone."

**Standard vocabularies** have been yet addressed as an useful tool for information integration, but they just supply a shared terminology, with some obvious drawbacks in terms of semantics.

Information integration via a common schema appears in principle the simplest way, but experience shows that this approach will almost invariably fail. The main reason is that different schemata exists as the heritage of well established cultural traditions and scholars are reluctant to accept a schema different from their own. It is frequent that there is a complete agreement on the semantics of some information elements, but it results impossible to reconcile two different data structures [Signore1993]. As a consequence, it is difficult, if not impossible at all, to agree on a single schema.

Integration is often attempted at metadata level. A typical example is the Dublin Core initiative\(^9\). In this approach, adopted in many projects, information is enriched by metadata, which permits to have a common reference schema. However, as also noticed in [Doerr2003], "the number of metadata vocabularies will continue to grow as individual communities seek to structure their own information for their own purposes", and "attempts to develop universal metadata vocabularies are misdirected, since "spoken" languages (those used by communities to actively describe content) will inevitably diverge (history is replete with failures to find common spoken languages [Eco1997])". In addition, in our opinion, metadata by themselves cannot exploit the full richness of possible associations among different information items. The association mechanism remains in the mind of the user.

A more useful effort is to attempt to formulate a language as a base for "understanding". This is what we can define to be a "core ontology" which incorporates basic entities and relationships common across the diverse metadata vocabularies.

### 3.1 What is an ontology

The term *ontology* was taken from philosophy, where it denotes a specific subfield, namely, the study of the nature of existence (the literal translation of the Greek word *Oντλoγια*). It is the branch of metaphysics concerned with identifying, in the most general terms, the kinds of things that actually exist, and how to describe them. The observation that the world is made up of specific objects

that can be grouped into abstract classes based on shared properties is a typical ontological commitment. More recently the term ontology has become relevant in the Knowledge Engineering community, taking a specific technical meaning, rather different from the original one. In fact, instead of "Ontology" we speak of "an ontology". Several different definition of ontology exist, highlighting different aspects. We will quote some of them [Gómez-Pérez2004]. According to [Neches1991]:

> An ontology defines the basic terms and relations comprising the *vocabulary* of a topic area as well as the *rules* for combining terms and relations to define *extensions* to the vocabulary.

We have to note that according to this definition, an ontology includes both the terms that are explicitly defined in it, as well as the knowledge that can be inferred from them.

One of the most popular definition of ontology was given by [Gruber1993]:

> An ontology is an explicit representation of a conceptualization.

and has been the base for several other definitions. [Borst1997] gave a slightly modified version:

> Ontologies are defined as a formal specification of a shared conceptualization.

and [Studer1998] merged the two in a longer and probably better explained definition:

> An ontology is a formal, explicit specification of a shared conceptualisation. A "conceptualisation" refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. "Explicit" means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal and a constraint is that a disease cannot cause itself. "Formal" refers to the fact that the ontology should be machine readable, which excludes natural language. "Shared" reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

We must quote two further definitions, which put emphasis on the formal aspects and logic. The first one [Guarino1995] goes a step further the Studer’s definition because formalizes the notion of conceptualization and establishes how to build the ontology by making a logical theory. According to this definition, which would be applicable only to ontologies developed in *logic*, an ontology is:

> A logical theory which gives an explicit, partial account of a conceptualization.
and has been further refined [Guarino1998] as:

A set of logical axioms designed to account for the intended meaning of a vocabulary.

For sake of completeness, we quote also the definition given in [Hendler2001], which is quite simple to understand and explain:

A set of knowledge terms, including the vocabulary, the semantic interconnections and some simple rules of inference and logic for some particular topic.

Sometimes the concept of ontology is somehow diluted, considering taxonomies as full ontologies, because they provide a consensual conceptualization of a domain. A distinction is made between lightweight and heavyweight ontologies. The former include concepts, concept taxonomies, relationships between concepts, and properties which describe concepts. The heavyweight ontologies add axioms and constraints, which clarify the intended meaning of the terms included in the ontology.

Even if there are many definitions of ontology, there is a true consensus among the ontology community, as the different definitions just provide different and complementary points of view. We can say that ontologies include terms explicitly defined and knowledge we can infer, and aim to capture consensual knowledge, to reuse and share across software applications and by group of people.

As far as the language used to implement them, we can classify them as highly informal (expressed in natural language, hence not machine readable), semi-informal (restricted and structured form of natural language), semi-formal (artificial and formally defined language, as OWL), and rigorously formal (when we have meticulously defined terms with formal semantics, theorem and proofs of properties).

### 3.2 Levels of knowledge representation

The degree of formalisation of concepts and their relations varies considerably between different domains of knowledge. At the lower end one finds lexicons and simple taxonomies, at the middle level one might place thesauri, at the high end of formalization of knowledge there are axiomatized logic theories. Such theories include rules to ensure the well-formedness and logical validity of statements expressed in the language of the scientific discipline [Digicult2003].

According to [Sheth2003] semi-formal ontologies, defined as ontologies that do not claim formal semantics and/or are populated with partial or incomplete knowledge, can be significantly smaller, especially for the ontology population effort, compared to that required for developing formal ontologies or ontologies with more expressive representations. Semi-formal ontologies have provided good examples of both value and utility in meeting several challenges; especially that of information integration. One key reason is that of the need to accommodate partial (incomplete) and possibly inconsistent information, especially in
the assertions of an ontology. Real world applications often can be developed with very little semantics ("little semantics goes a long way" [Hendler2006]), or with compromises with completeness and consistency required by more formal representations and inferencing techniques.

Hierarchical classification systems and structured vocabularies do not lend themselves easily to rich inter-linking of conceptual "trees". A major step further in this direction is the "CIDOC object-oriented Conceptual Reference Model" (CRM). This provides an ontology of 81 classes and 132 unique properties, which describes in a formal language concepts and relations relevant to the documentation of cultural heritage.\footnote{The CIDOC CRM has been accepted as working draft by ISO/TC46/SC4/WG9 in September 2000. Since 9/12/2006 it is official standard ISO 21127:2006. See http://cidoc.ics.forth.gr/ for details.}

Even if both a core ontology and core metadata, such as Dublin Core, are intended for information integration, they differ in the relative importance of human understandability. Metadata is in general thought for human processing, while a core ontology is a formal model for automated tools that integrate source data and perform a variety of functions. Vocabularies based on ontologies that organize the terms in form that has a clear and explicit semantics can be reasoned over, which is a fundamental process in enriching knowledge, inferring new information about resources. CIDOC CRM is a formal ontology for cultural heritage information specifically intended to cover contextual information. It can be used to perform reasoning (e.g. spatial, temporal).

4 Semantic Web technologies

4.1 The Semantic Web Stack

According to the vision by Tim Berners-Lee, the Web has a layered architecture, which is under development since several years (see figure 1 for the most recent version of this stack).

To understand the framework, we recall that the Web must be seen as a Universal Information Space, navigable, with a mapping from URI (Uniform Resource Identifier) to resources. In this framework, the adjective semantic means "machine processable". The Semantic Web, much like XML, is a declarative environment, where we specify the meaning of data. For the semantic web to function, it is needed that computers have access to structured collection of information and set of inference rules that they can use to conduct automated reasoning. The challenge of the semantic web is therefore to provide a language that expresses both data and rules for reasoning about data and that allows rules from any existing knowledge-representation system to be exported onto the Web.

Semantic Web is a "hot" research topic, and many applications are emerging, both in academia as well at the industrial level. In the following we will briefly describe some of the Semantic web stack components.
4.2 XML and XML Schema

Extensible Markup Language (XML) was born to overcome HTML limitations in implementing new data-centric Web applications. Therefore it was a first step to supply semantics to tags and support web transactions, allowing exchange of information among different databases. Great advantages are the possibility of having different views of the same data, and personalize their presentation by means of appropriate agents. Finally, it must be stressed that XML, independent from platforms and languages, plays a fundamental role toward interoperability. According to XML syntax, every information field, called element, must be enclosed between a start and end tag (e.g. `<aut>` and `</aut>`). An element can also have attributes, made by name/value pairs. Tags must be correctly nested, we can have empty elements, attribute values must be enclosed in quotes. Syntax is very simple, automatic processing is easy, and source docu-
ments remains human readable. XML is extensible and flexible, and is the basic technology adopted to model the web. All new languages are described using XML.

We can have a formal description of the otherwise implicit document structure, named DTD (Document Type Definition), that can be included in the document itself, or can be referenced as an external resource. A XML document is "well formed" if it matches writing rules, is "validated" when its structure is coherent with the referenced DTD.

DTDs are expressed using their own syntax; therefore they require appropriate editors, parsers, processors. In addition, their extension is not easy, datatype concept is missing. Finally, DTD must support all elements and attributes described in the included namespaces. Schemas play a role similar to DTDs, but offer significant advantages: are written according to XML syntax, include datatypes, inheritance, schema combination rules, offer better namespace support and allow linking of semantic information. Using XMLSchema, designer can specify value constraints, complex types and type hierarchies.

4.3 RDF

The foundation of Resource Description Framework (RDF\(^\text{11}\)) is a model for representing named properties and property values. The RDF model draws on well-established principles from various data representation communities. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources and an RDF model can therefore resemble an entity-relationship diagram. The RDF data model is a syntax-neutral way of representing RDF expressions. The basic data model consists of three object types:

**Resources** All things being described by RDF expressions are called resources. A resource may be an entire Web page; or be a part of it, (e.g. a specific HTML or XML element within the document source). A resource may also be a whole collection of pages (e.g. an entire Web site) or be an object that is not directly accessible via the Web (e.g. a book, a painting, etc.) Resources are always named by URIs plus optional anchor ids. Anything can have a URI; the extensibility of URIs allows the introduction of identifiers for any entity imaginable.

**Properties** A property is a specific aspect, characteristic, attribute, or relation used to describe a resource. Each property has a specific meaning, defines its permitted values, the types of resources it can describe, and its relationship with other properties. Each property is identified by a name, and takes some values.

\(^{11}\)See http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ for an introduction and reference to other documents.
**Statements** A specific resource together with a named property plus the value of that property for that resource is an RDF statement. These three individual parts of a statement are called, respectively, the subject, the predicate, and the object. The object of a statement (i.e., the property value) can be another resource or it can be a literal; i.e., a resource (specified by a URI) or a simple string or other primitive datatype defined by XML. A set of properties referring the same resource is called description.

We can diagram an RDF statement pictorially using directed labeled graphs (also called "nodes and arcs diagrams"). In these diagrams, the nodes (drawn as ovals) represent resources and arcs represent named properties. Nodes that represent string literals will be drawn as rectangles.

### 4.4 RDF Schema

RDF is an universal language allowing users to use their own vocabularies to describe resources. As a consequence, RDF does not make assumptions about any particular application domain, nor does it define the semantics of any domain. It is up to the users to do so in RDF Schema (RDFS). The main issue is that in many cases we have to talk not only about individual objects (also known as resources), but about classes that define types of objects. In fact, a class can be thought as a set of elements, and the individual elements belonging to a class are referred as instances of the class. In RDF we can use `rdf:type` to specify the relationship between instances and classes. Classes can be used to impose restrictions on what can be stated in an RDF document using that schema. For example, we would state that propositions like:

```
Leonardo isAuthorOf Gioconda.
Cimabue wasMasterOf Giotto
```

have a sense, while statements like

```
Michelangelo isAuthorOf Leonardo.
Portrait of Julius the 2nd isAuthorOf Gioconda.
```

are both nonsensical, because an artist can’t be author of an artist, and an artifact can’t be the author of an artifact. However an artist could be the master of another artist, and an artifact can be a different version (or a copy of) another artifact. More formally, we can say that we would need to restrict respectively the domain or the range of the property `isAuthorOf`.

Once we have classes, we would like to establish relationships between them. For example, suppose we defined the following classes: person, artist, artifact, painting, statue. It is easily seen that these classes are related. In particular, every artist is a person (or, more formally, artist is subclass of person, and person is superclass of artist). Similarly, painting and statue are both subclass of artifact. This kind of relationship between classes is known as hierarchy of classes, which must not necessary be a strict hierarchy, meaning that a class can
have multiple superclasses. When a class has more than one superclass, it just means that every instance of that class is at the same time an instance of all its superclasses.

A very important consequence of hierarchical organization of classes is the inheritance. This means that, for example, if we restrict the domain and the range of the property isAuthorOf, saying that the domain must be a person, and the range must be an artifact, we will have as a consequence that an artist can be author of a painting or a statue, because the elements of the class artist inherit the properties of the class person, and the classes painting and statue inherit the properties of their superclass artifact.

In other words, RDF Schema fixes the semantics of "is subclass of", whose intended meaning will be used directly by all the RDF processing software, with no need to be interpreted by the application. RDF Schema can therefore be used to define, to a limited extent, the semantics of a particular domain, and can be seen as a (primitive) ontology language.

In RDFS properties are defined globally, and are not encapsulated as attributes in class definition. Therefore we can define new properties that apply to existing classes, making possible to use classes defined by others, adapting them to our requirements just adding new properties.

We can also define hierarchical relationships between properties. For example, painterOf is a subproperty of isAuthorOf.

The W3C's Resource Description Framework (RDF) Schema Specification 1.0, in a section on its scope, mentions concept navigation, and states: "Thesauri and library classification schemes are well known examples of hierarchical systems for representing subject taxonomies in terms of the relationships between named concepts. The RDF Schema specification provides sufficient resources for creating RDF models that represent the logical structure of thesauri (and other library classification systems)."

### 4.5 OWL

RDF and RDF Schema have a limited expressivity, as RDF is roughly limited to binary ground predicates, and RDF Schema is limited to subclass and property hierarchy, with domain and range definitions of these properties. Advanced applications will require much more expressiveness, to write explicit and formal conceptualizations of domain models. Among the others, we will recall: a well-defined syntax, a formal semantics, sufficient expressive power, efficient reasoning support and convenience of expression [Antoniou2004].

The W3C Web Ontology Working Group, starting from previous projects (DAML + OIL), ended in the definition of OWL (Web Ontology Language).

We will not discuss here in detail the requirements for an ontology language, limiting ourselves to point that the syntax of both DAML+OIL and OWL, built upon RDF and RDFS, is not user friendly, but this is of little concern for users, which will normally use ontology development tools, while much more important is the use of a formal semantics allowing people to reason about the knowledge, so checking consistency of the ontology and the knowledge, discovering possible
unintended relationships between classes, or automatically classifying instances in classes. Usually formal semantics and reasoning support are provided mapping an ontology language to a known logical formalism, and by using automated reasoners for those formalisms. In the case of OWL, we have a partial mapping on a description logic (a subset of predicate logic) for which efficient reasoning support is possible, and several reasoners exist.

A good ontology language should have several sophisticated features, however, the more expressive will be the language, the less efficient will be the reasoning, and, especially defining an ontology language which will operate at the web level, we need a compromise between expressiveness and computability. From these considerations we got the three flavors of OWL:

OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. Provides a quick migration path for thesauri and other taxonomies. Owl Lite also has a lower formal complexity than OWL DL. It does not support constructs like arbitrary cardinality, disjointness statements and enumerated classes.

OWL DL (OWL Description Logic) supports those users who want the maximum expressiveness while retaining the guarantee that all conclusions are computable (computational completeness) and all computations will finish in finite time (decidability). OWL DL permits efficient reasoning support, but it is not fully compatible with RDF. To get a legal OWL DL document from an RDF one, we have to restrict it in some ways and extend in other. However, any legal OWL DL document is a legal RDF document.

OWL Full for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. Any legal RDF document is a legal OWL Full document, because OWL Full is is fully upward compatible with RDF, both syntactically and semantically. However, the language is undecidable, and there is no complete or efficient reasoning support.

4.6 An example

In the following example we report a fragment of CIDOC CRM ontology, populated with two instances ("Michelangelo Buonarroti" and "painting of Sistine Chapel") modeling the simple fact that: the painting of the Sistine Chapel (E7) was carried out by (P14F) Michelangelo Buonarroti (E21).

The reader will note as the relation (property) P14F.carried_out_by is defined as having E7.Activity as range and E39.Actor as domain, and E21.Person is a subclass of E39.Actor.

Needless to say, the coding is unsuitable for human reading, but formally well defined and understandable by machines.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [ [
```
<!-- Ontology Information -->
<owl:Ontology rdf:about=""
owl:versionInfo="3.4.9">
</owl:Ontology>

<!-- Classes -->
<owl:Class rdf:about="#E1.CRM_Entity">
  <rdfs:label xml:lang="en">E1.CRM Entity</rdfs:label>
</owl:Class>

<owl:Class rdf:about="#E2.Temporal_Entity">
  <rdfs:label xml:lang="en">E2.Temporal Entity</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E1.CRM_Entity"/>
  <owl:disjointWith rdf:resource="#E77.Persistent_Item"/>
</owl:Class>

<owl:Class rdf:about="#E21.Person">
  <rdfs:subClassOf rdf:resource="#E39.Actor"/>
</owl:Class>

<owl:Class rdf:about="#E39.Actor">
  <rdfs:subClassOf rdf:resource="#E77.Persistent_Item"/>
</owl:Class>

<owl:Class rdf:about="#E4.Period"/>
<rdfs:label xml:lang="en">E4.Period</rdfs:label>
<rdfs:subClassOf rdf:resource="#E2.Temporal_Entity"/>
</owl:Class>

<owl:Class rdf:about="#E5.Event">
<rdfs:label xml:lang="en">E5.Event</rdfs:label>
<rdfs:subClassOf rdf:resource="#E4.Period"/>
</owl:Class>

<owl:Class rdf:about="#E7.Activity">
<rdfs:label xml:lang="en">E7.Activity</rdfs:label>
<rdfs:subClassOf rdf:resource="#E5.Event"/>
</owl:Class>

<owl:Class rdf:about="#E77.Persistent_Item">
<rdfs:label xml:lang="en">E77.Persistent Item</rdfs:label>
<rdfs:subClassOf rdf:resource="#E1.CRM_Entity"/>
<owl:disjointWith rdf:resource="#E2.Temporal_Entity"/>
</owl:Class>

<!-- Annotation Properties -->
<owl:AnnotationProperty rdf:about="&rdfs;comment"/>
<owl:AnnotationProperty rdf:about="&rdfs;label"/>
<owl:AnnotationProperty rdf:about="&owl;versionInfo"/>

<!-- Object Properties -->
<owl:ObjectProperty rdf:about="#P11B.participated_in">
<rdfs:domain rdf:resource="#E39.Actor"/>
<rdfs:label xml:lang="en">P11B.participated in</rdfs:label>
<rdfs:range rdf:resource="#E5.Event"/>
<owl:inverseOf rdf:resource="#P11F.had_participant"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#P11F.had_participant">
<rdfs:domain rdf:resource="#E5.Event"/>
<rdfs:label xml:lang="en">P11F.had participant</rdfs:label>
<rdfs:range rdf:resource="#E39.Actor"/>
<owl:inverseOf rdf:resource="#P11B.participated_in"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#P12B.was_present_at">
<rdfs:domain rdf:resource="#E77.Persistent_Item"/>
<rdfs:label xml:lang="en">P12B.was present at</rdfs:label>
<rdfs:range rdf:resource="#E5.Event"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#P12F.occurred_in_the_presence_of">
<rdfs:domain rdf:resource="#E39.Actor"/>
<rdfs:label xml:lang="en">P12F.occurred in the presence of</rdfs:label>
<owl:inverseOf rdf:resource="#P11B.participated_in"/>
</owl:ObjectProperty>
5 Why an ontological approach?

In the past years a big emphasis has been put on XML data structuring, but XML is semantically poor. The Semantic Web stack higher levels technologies (RDF, OWL, etc.) can supply the appropriate technical environment to represent, export and share the knowledge needed to implement intelligent retrieval and browsing systems, and reason upon data. In the peer-to-peer web archi-
tecture, Semantic Web technologies allow fully decentralized semantic markup of content (for example, using classes and properties defined in CIDOC-CRM). Intelligent software agents can then use knowledge expressed by the markup.

Looking back to the history of data cataloging and sharing of cultural heritage information, we can see how we progressed from initial stages, where info was entered in an informal way, to more structured organization of information, and now we have many projects referring to a common metadata set (mainly Dublin Core, sometimes Qualified Dublin Core). Some more advanced projects [Hyvönen2004] rely on ontologies, mainly as set of related terms to use for more precise queries.

The question, looking at the common agreement upon the metadata set, is why we should consider an ontological approach? Reasons are essentially related to the implicit assumption made with the metadata approach. In short, adding metadata to the description of an artifact (an instance of the class artifact), implicitly means that we assume a one-to-many (or possibly many-to-many) relationship between the artifact and the classes identified by the metadata. As an example, specifying some DC metadata like:

```xml
dc:title=Pietà
dc:creator=Michelangelo
dc:date=1499
dc:subject=Madonna
dc:subject=Christ
```

or

```xml
dc:title=Madonna del cardellino
dc:creator=Raffaello
dc:date=1505
dc:subject=Madonna
dc:subject=Child
```

we intend to say that both "Michelangelo" and "Raffaello" are instances of the class artist, and that a particular artifact (the Pietà, for example) has been made by Michelangelo, is dated 1499, and has subject "Madonna" and "Christ", while the second one (the painting) has been made by Raffaello, is dated 1505, and has subject "Madonna" and "Child".

We can add controlled vocabularies to be sure that we specify correct terms for "creator" or "subject", but only humans can:

- check the consistency between `dc:creator` and `dc:date` as no artifact can be made by an artist after her/his death, or before her/his birth date (plus, let us say, 10 years?);

\[\text{In the following examples, for sake of simplicity we are not conforming to a true and valid syntax, which would require expressions like:}
<\text{meta name}=\text{dc.creator} \text{content}=\text{Michelangelo} /> \text{or}
<\text{dc:creator}>\text{Michelangelo}</dc:creator>\]
• having found an artifact, search for artifacts made in the same period, or by artists which were living and active in the same period;

• find artifacts (for example portraits) which are "imaginary" portraits, because the scene is imaginary, or subjects never existed because they are mythological, or subjects were not existing at the time of life of the artist or in the same time themselves (for example the famous painting "School of Athens" by Raffaello).

Available thesauri are supposed to support these needs, but thesauri are often designed aiming to more effective retrieval, instead of formally representing the knowledge, and can’t be automatically translated into ontologies, as they sometime model a class-subclass relationship (like "statues" and "korai (statues)"), sometime model just different instances (for example, "Renaissance" is sometime modeled as a BT of "15th century", while both are period in time, having some spanning). Multiple inheritance and time dependent relationships are also an issue.

6 A scenario

Much information is conveyed by the links connecting different pieces of information. Web searching and browsing can take advantage of the interoperable knowledge representation to appropriately link information following the user preferred interaction metaphors (spatial, temporal, classification affinity), so greatly improving the access to information and knowledge stored in museums.

In the semantic web environment intelligent user agents can rely on a core ontology to understand the mental model expressing the user interests, implementing suitable navigation mechanisms.

6.1 (Semantic) Searching the Web

Most of the value in browsing the web comes from following associations (links) coherent with user’s scientific interests. Links have their own semantics. This aspect has been often neglected, even if it was present since the inception of the Web, as we can easily see from the original proposal by Tim Bernes-Lee, and is the key for really supporting the association model that is the basis of the hypertext.

In the Web context, documents, whatever will be their genesis, are seen as resources. We can model the association process as a sequence of steps where the anchor leads to a concept, the concept is related to other concepts, the new concept is related to some resources. In the data space, documents are connected by extensional links. In the Semantic Web architecture’s ontological level, associations among concepts implement intensional links among documents. It is easy to realize as important are the links, just considering the typical "search and link" approach followed by the majority of users. In this typical usage scenario, the user starts with a fairly general query, and the search engine often
returns a huge amount of records. The user then looks for the most “promising” records. Once an interesting one has been found, (s)he tends to follow the links, so making use of the knowledge embedded in the document itself, as typically links are inserted by the designer to point to relevant connected information. Adaptive and intelligent systems should take care that navigation is consistent with the real user interests, that could be formalized by a representation of the user’s mental model.

The traditional Information Retrieval approach, based on term matching (sometimes enriched with smart ranking algorithms) is essentially a mere syntactic approach, where the system returns records containing the words specified by the user, while users prefer formulate queries using high level semantic concepts, more consistent with standards and tacit knowledge. In contrast, the semantic search [Guha2003] is an application of Semantic Web to search, supporting Research Searches where the user provides the search engine with a phrase which is intended to denote an object about which the (s)he is trying to gather/research information. There is no particular document which the user knows about that (s)he is trying to get to. Rather, the user is trying to locate a number of documents which together will give him/her the information (s)he is trying to find. [Guha2003] points as ontologies can give better results due to the availability of machine understandable structured knowledge and better identification of concepts to search for. We can then enrich the result list, and get higher quality information thanks to the text understanding and processing supported by ontologies.

6.2 Ontology driven access to museum information

We can imagine [Signore2005] an architecture where intelligent user agents can have access to the mental model expressing the interests of the user. The content can be tagged and semantically annotated using classes and properties defined in CIDOC-CRM. The agent can then perform reasoning, linking the information the user is interested in, following the relevant associations.

The user mental model can be expressed in terms of preferred interaction metaphors. Making reference to the ontology used as basis for semantic annotation, this means to specify the set of classes and properties the user can be interested in navigating. A user interested in the temporal context will be interested in classes like: E2.Temporal_Entity, E52.Time-span and their subclasses, at various levels, like E3 Condition State, E4.Period, E5.Event. The context can be expressed in a more precise way stating the properties the user is interested to navigate (e.g. P117.occurs during, P118.overlaps in time with, etc.) to build up the temporal interaction metaphor. Identifying the properties the user is interested in can guide the agent to select the appropriate associations and perform the reasoning, for example, to supply information to the user interested in knowing: what was happening in the period 1498-1505?

The user agent (the browser) is enriched by two components: a reasoner and a finder, which accomplish the tasks of getting the semantic annotation of the current resource, looking to the user model, finding correspondences between
user model and resource metadata, initiating a search following the properties
the user is interested in.
Suppose for example that the resource is describing a painting done in 1530,
describing an event pertaining to the history of Christ, by a painter of Sicilian
school, and the user is interested in the temporal context, the reasoner can
follow the properties relating year 1530 to historical or artistic period, or events
occurred in a suitable time interval around 1530.

7 Conclusion

Cultural heritage applications are a challenge, where many conceptual and im-
plementation issues date since many years. A lot of effort has been paid to the
implementation of appropriate user aids, looking for information integration,
which remains a prerequisite to contextualize objects and understand the cul-
ture that is behind them. However, all these aids are thought for human usage,
and can’t be understood by machines, neither can support automated reason-
ing. All attempts to information integration based upon common schemata
have been a failure, and integration at metadata level cannot exploit the full
richness of possible associations among different information items. We need a
formal, explicit specification of a shared conceptualization (hence an ontology)
to support contextual information and perform automated reasoning.

Semantic Web technologies can support the explicit, machine understandable
representation of knowledge, which can be exported and shared among different
people and applications. It is important to stress how the distributed nature of
the Web allows to describe properties of objects stored anywhere in the world.

The ontological approach differs from the usual ones, based on cataloging
cards, common schema, shared metadata vocabularies, because it goes behind
the implicit "flat" resulting schema, can represent the richness of interconnec-
tions among different information items, without imposing any centralization,
and can support automated reasoning. If the knowledge accumulated in decades
by scholars will be made available and shared, using Semantic Web technologies,
any user will be able to search the online universe seamlessly, as if (s)he could
access one vast library of information, and intelligent user agents could under-
stand the user’s mental model, to support different and appropriate interaction
metaphors.

References

Primer, The MIT Press, Cambridge, Massachusetts, April

for Telematica and Information Technology, University of


