Integrating Ontologies and Thesauri to Build RDF Schemas

Bernd Amann[1] and Irini Fundulaki[2]

[1] Cedric CNAM, 292 Rue St. Martin, 75141 Paris Cedex 03 France
email : amann@cnam.fr
[2] INRIA Rocquencourt, 78153 Le Chesnay Cedex, France
email : irini.fundulaki@inria.fr

Abstract. In this paper we present a new approach for building RDF schemas by integrating existing ontologies and structured vocabularies (thesauri). We will present a simple mechanism based on the specification of inclusion relationships between thesaurus terms and ontology concepts and show how these relationships can be exploited to create application-specific RDF schemas incorporating the structural views of ontologies and deep classification schemes provided by thesauri.

1 Introduction

With the emergence of the World Wide Web, Internet and Intranet technologies, a large number of information sources from a variety of different application domains have become available on line. In such open and evolving environments, discovering, accessing and integrating information are difficult and complex tasks due to the existence of *semantic heterogeneities* [35], resulting from the different terminologies and conceptualizations employed by the various information providers and consumers.

A partial solution to the semantic heterogeneity problem is the exchange of *domain-specific metadata* [22, 41, 35] between interconnected systems, describing the semantics of the underlying information. More specifically, these semantics are expressed by *metadata schemas*, defined by specific resource description communities. A metadata schema is comprised of (1) a vocabulary, i.e. a set of element names to be used for the description of information in a domain (e.g. the *creator*, *title* elements of the Dublin Core [12] metadata element set), and (2) a set of semantic relationships to structure this information. One of the several roles of metadata schemas in open and evolving environments such as the Web, is to support a sharable, structural view of information with rich semantics to be communicated between users and applications.

Metadata specification languages, such as the *Resource Description Framework* (RDF) [34, 6], support standard mechanisms for the representation of metadata schemas as well as source specific metadata (source descriptions). RDF is an ongoing standardization effort of the World-Wide Web Consortium (W3C) for the creation of metadata describing Web resources. Although it enables the description and exchange of metadata schemas, it does not provide a mechanism to facilitate their construction, which is a difficult and time consuming task especially in environments that comprise a large number of information sources. Moreover, it offers no mechanism to decide whether a particular metadata schema meets the needs of an application or domain. For that sake, we need to consider semantic components and structural views that describe the organization of the underlying information.

In this paper, we present a modular approach for the creation of RDF *schemas* based on the integration of existing *ontologies* and *thesaurus hierarchies* defined according to the ISO 2788 [20] standard for monolingual thesauri. Ontologies and thesauri can be considered as orthogonal ways for describing information. The former provide structural, sharable views of information, with usually shallow semantics, captured in metadata schemas. They are declarative specifications of the *concepts* and *roles* in a domain of discourse. Thesauri are structured vocabularies, with rich semantics but little or no structure. For example, although the *Art & Architecture Thesaurus*, one of the largest thesauri in the field of western art terminology, includes extended taxonomies of cultural artifacts and styles, there is no explicit relationship denoting the fact that artifacts have a style. In the context of our approach, ontologies are perceived to have a dual role: provide a generic view of information and a structural interface over thesauri.

We follow a three-step approach to the construction of RDF schemas. In a first step, we specify for each thesaurus term, a set of ontology concepts, the former being considered as sub-concepts of the latter. The result of this step is a *connection relation* between terms and concepts with inclusion semantics. In a second, intermediate step, we extract *automatically* for each concept a *concept thesaurus*. This thesaurus contains only the terms connected to this concept by the connection relation, along with *broader-generic* relationships derived from the initial thesaurus. In the final step we integrate these thesauri with the ontology to produce an RDF schema consisting of (1) a *structural view* provided by the ontology, (2) *connection relations* between concepts and terms, and (3) *thesaurus hierarchies*. With this intermediate step it is possible to construct the resulting schema incrementally by extracting on demand concept thesauri that correspond to different ontology concepts.

Our contribution is two-fold. First, by using existing components, we minimize the time and effort to specify appropriate notions that describe the content and structure of a domain in the form of an RDF schema. Second, the resulting RDF schema is not bound to a specific implementation and can be used by any application which is based on the RDF standard.

To illustrate our approach, we take examples from the cultural application domain. Thesaurus examples are taken from the *Art & Architecture Thesaurus* (AAT). The *Art & Architecture Thesaurus* is one of the Getty Information Institute's (http://www.gii.getty.edu/) ongoing projects and known as one of the largest thesauri in the area of western art historical terminology. Ontology examples are inspired from the *ICOM/CIDOC* Reference Model. The *ICOM/CIDOC*

Reference Model [19] is the result of one of the most significant efforts for a formal representation of the basic notions of the cultural application domain.

This paper is organized as follows. Related work is presented in Section 2. Section 3 gives a short presentation of RDF. In Sections 4 and 5, we describe the notion of ontology and thesaurus respectively. In Section 6, we present our approach to the automatic construction of RDF schemas by integrating ontologies and thesaurus hierarchies. Conclusions and future work are given in Section 7.

2 Related Work

Over the past years a great amount of effort has been invested in the development of metadata vocabularies for the exchange of information across different applications and domains [12, 25, 40, 10]. Dublin Core [12] contributes to semantic interoperability by promoting a common set of elements which can be used to describe in a consistent manner information concerning the contents of electronic documents, such as their *title*, *creator*, or *subject*. USMARC [40] defines a set of descriptive elements for the representation and exchange of bibliographic data. In the cultural domain, the Aquarelle Project [31] uses the SGML CI DTD (Data Type Definition) of the French Ministry of Culture [10] to describe a set of element names, dedicated to territory inventory making. All the above metadata element sets are the result of the collaboration of a number of user communities and other authorities in the corresponding fields. Our approach can be considered as a methodology to provide such metadata element sets by using existing semantic components of the domain of interest, namely ontologies and thesaurus hierarchies.

Besides specific metadata element sets, ontologies have been developed and used in several projects to structure and access Web knowledge. The OntoSeek [18] system is used for gathering and organizing Web source descriptions. It exploits the SENSUS Ontology [24] which is based on the WordNet [32] linguistic ontology to describe source contents. The WebKB set of tools [29] builds on a terminological ontology and conceptual graphs to represent (and index) documents. Our approach can be considered complementary to the above systems, in the sense that not only do we provide a methodology to define ontologies enriched with thesaurus hierarchies, but also the choice of RDF as the representation language enables their exchange in a machine readable format.

Besides structuring and representing Web data, metadata schemas (referred to as *domain models*) are also used in mediation based systems such as Information Manifold [4], SIMS [9] and Carnot [11]. They provide a uniform view of information in a domain of discourse and are used to describe the contents of different information sources. For example, Carnot is an information integration system that relies on the CYC [28] knowledge base for describing source contents. The CYC knowledge base is a formalized representation of a *"vast quantity of fundamental human knowledge"* and contains about 10^5 general concepts and 10^6 assertions on these concepts. We do not aim at providing a mediation system but rather a methodology to define mediator domain models. The interesting issue is that in some cases, underlying sources might use thesaurus hierarchies that could be integrated in the mediator to produce expressive domain models. Our approach can be considered as a first step towards this integration, which is a requirement for the next generation information systems [35, 33].

Integrating ontologies and thesauri can be considered as a schema integration problem [5]. An important issue in this field concerns the coherent integration of database schemas with overlapping concepts, roles and data. Our approach is more simple since ontologies and thesauri can be considered as orthogonal ways of describing information. First, ontologies capture more general semantics than thesauri, and consequently we consider thesaurus terms to be specializations of ontology concepts. Second, thesauri incorporate only a fixed set of *semantic relationships* defined independently of any application or domain. Finally, the consistency of the resulting metaschema is not based on actual data, but on the meaning of ontologies and thesauri as perceived by experts in the domain.

3 Resource Description Framework

The *Resource Description Framework* (RDF) is a *foundation for processing metadata* [34, 6] which supports standard mechanisms for the representation of metadata schemas as well as source descriptions. It relies on a simple, graph-based data model and uses XML (eXtensible Markup Language) [42], to communicate and process metadata in a machine readable and human understandable format. Similar to the separation of schema and instance in traditional databases, we can distinguish between RDF descriptions and RDF schemas, the former considered as instances of the latter.

3.1 RDF descriptions

RDF can be used to describe any kind of *resource* [26] that is identified by a URI (Uniform Resource Identifier), such as a Web server, an XML document or an element of an HTML page (e.g. an image). RDF supports the definition of resource *properties* whose values can be other resources or literals (strings, integers). A collection of *property/value* pairs that refers to a specific resource is called an RDF *description* and can be represented as a labeled directed graph where nodes correspond to resources or literals (values) and edges to resource properties.

Figure 1 shows an RDF description for a Web page that describes a painting of the French painter Claude Monet. RDF uses the XML *namespace* mechanism to distinguish among different RDF schemas (Section 3.2) used in RDF descriptions. For example, lines `2` and `3` define two XML namespaces where the first (`web-page`) contains general properties of HTML pages (`title, presents, creator`) and the second (`artifact`) specifies properties of cultural artifacts (`title, style, type, period`). This mechanism is very important since it permits the reuse of existing, distinct RDF schemas within the same RDF description, without creating naming conflicts (e.g. `web-page:title`, `artifact:title`).

Line 4 tells us that the description that follows concerns the HTML page which can be accessed by the URL `http://metalab.unc.edu/louvre/paint/monet-/first/impression/`. The title of this page is *"Web Museum: Monet, Claude : Impression : soleil levant "* (line 5) and has been created by Nicolas Pioch (line 14). To describe properties of the painting, it is necessary to define a local resource which is identified by URI `soleil_levant` that refers to the painting. The painting's properties are its type (`oil painting`, line 8), title (`Impression : soleil levant`, line 9), style (`impressionism`, line 10) and period (`first-impressionism`, line 11).

```
1. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2.   xmlns:web-page ="http://metalab.unc.edu/louvre/namespaces/web-pages"
3.   xmlns:artifact ="http://metalab.unc.edu/louvre/namespaces/artifacts">
4.  <rdf:Description
      about="http://metalab.unc.edu/louvre/paint/monet/first/impression">
5.    <web-page:title>Web Museum: Monet, Claude: Impression: soleil levant
      </web-page:title>
6.    <web-page:presents>
7.      <rdf:Description about="soleil_levant">
8.        <artifact:type>oil painting</artifact:type>
9.        <artifact:title>Impression : soleil levant</artifact:title>
10.       <artifact:style>impressionism</artifact:style>
11.       <artifact:period>first-impressionism</artifact:period>
12.     </rdf:Description>
13.   </web-page:presents>
14.   <web-page:creator>Nicolas Pioch</web-page:creator>
15.  </rdf:Description>
16. </rdf:RDF>
```

Fig. 1. An RDF description for resource `http://metalab.unc.edu/louvre/paint/-monet/first/impression`.

3.2 RDF Schemas

The RDF Schema Specification Language [7] is a declarative language used for the definition of RDF schemas[1] incorporating aspects from knowledge representation models (e.g. semantic nets), database schema definition languages and graph models. It is a simple language of restricted expressive power compared to predicate calculus based specification languages such as CycL [28] and KIF [23].

An RDF schema defines classes and properties which can be instantiated in RDF descriptions. More specifically, an RDF schema is comprised of (1) a *vocabulary*, i.e. a set of class and property names to describe information in a

[1] In the following, *RDF Schema* will denote the specification language used to define RDF schemas.

domain (as for example, the *creator*, *title* elements of the Dublin Core metadata element set), and (2) a set of *semantic relationships* to structure this information. Classes are organized in hierarchies using the property `rdfs:subclassOf` which is defined in *RDF Schema* (namespace `rdfs`) and has the standard semantics of inheritance relationship in object-oriented data models. For example, the RDF schema illustrated in Figure 2 defines class `Man Made Object` (line 4) and its subclass `Iconographic Object` (lines 5,6). It also defines classes `Style` (line 7), `Period` (line 8). RDF Schema allows both typed and untyped properties. Properties in our example are typed (i.e. they have a restricted domain and range). In Figure 2, property `period` (line 15) is defined between classes `Man Made Object` (line 16) and `Period` (line 17), using the RDF Schema properties `rdfs:domain` and `rdfs:range` respectively.

Summarizing, RDF offers a rich, comparatively simple graph-based data model and supports the definition of source specific metadata (RDF descriptions) and metadata schemata (RDF schemas). It uses XML for the syntactical representation, exchange, and processing of these metadata.

```
1. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2.    xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
3.    xmlns:artifact="">
4.   <rdfs:Class rdf:ID="Man Made Object"></rdfs:Class>
5.   <rdfs:Class rdf:ID="Iconographic Object">
6.     <rdfs:subclassOf rdf:resource="#Man Made Object"/></rdfs:Class>
7.   <rdfs:Class rdf:ID="Style"></rdfs:Class>
8.   <rdfs:Class rdf:ID="Period"></rdfs:Class>
9.   <rdf:Property rdf:ID="style">
10.    <rdfs:domain rdf:resource="#Iconographic Object"/>
11.    <rdfs:range rdf:resource="#Style"/> </rdf:Property>
12.  <rdf:Property ID="title">
13.    <rdfs:domain rdf:resource="#Man Made Object"/>
14.    <rdfs:range  rdf:resource="#rdfs:Literal"/></rdf:Property>
15.  <rdf:Property rdf:ID="period">
16.    <rdfs:domain rdf:resource="#Man Made Object"/>
17.    <rdfs:range rdf:resource="#Period"/></rdf:Property>
18. </rdf:RDF>
```

Fig. 2. An RDF schema for describing cultural resources.

4 Ontologies

The term *ontology* has been used in several disciplines, from philosophy, to knowledge engineering, where an ontology is considered as a computational entity, containing concepts and their properties, relationships between concepts and constraints. Ontologies are defined independently of the actual data [16], reflect a common understanding of the semantics of the domain of discourse and are used to share and exchange semantic information between sources [15, 33]. They are declarative specifications of the basic concepts and roles in an application domain. We only consider ontologies with inheritance relations (isa) and typed roles between concepts, sufficient to model a large class of ontologies [17] that can be easily represented as RDF schemas (Section 3.2).

Definition 1. *An ontology is a triple* $\mathcal{O} = (C,\ R,\ isa)$ *defined as follows :*

1. $C = \{c_1, c_2, \ldots, c_n\}$ *is a set of concepts, where each concept* c_i *refers to a set of real world objects (concept instances),*
2. $R = \{r_1, r_2, \ldots, r_m\}$ *is a set of binary typed roles between concepts,*
3. isa *is a set of inheritance relationships defined between concepts. Inheritance relationships carry subset semantics and define a partial order over concepts.*

Ontologies can be represented as directed graphs where nodes correspond to concepts and arcs correspond to roles and isa relationships. Figure 3 illustrates an example ontology, inspired from the ICOM/CIDOC Reference Model [19] which is used to describe cultural information. Concept `Physical Object` collects all physical objects, the latter *composed of* other physical objects. Activities (concept `Activity`) are *associated with* physical objects, the former *performed by* persons, institutions and organizations (concept `Actor`). Concepts `Biological Object` and `Man-Made Object` are *sub-concepts* of `Physical Object` and inherit all roles defined in their superclass. Instances of `Man-Made Object` have a title (role *has-title*) and have been created in a specific period (role *of-period*). `Iconographic Object` is a sub-concept of `Man-Made Object`. Iconographic objects have a style (role *style*) which is an instance of concept `Style`.

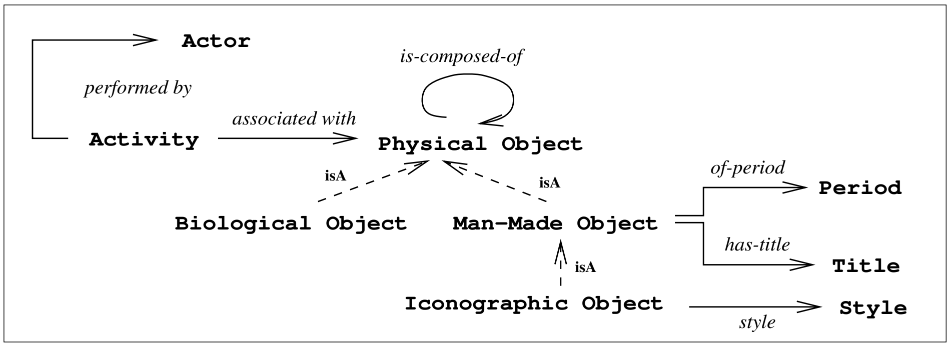

Fig. 3. A simple cultural ontology.

5 Thesauri : Structured Vocabularies

A vocabulary is a collection of *terms* that describe information in a domain of interest. Examples of such vocabularies are the *ACM Computing Classification System* [2], the Library of Congress Subject Headings [27], the *Unified Medical Language System* [39] for medicine and the *Art & Architecture Thesaurus* [38, 1] for the cultural domain. Thesauri are structured vocabularies of thousands of terms which have and are being used as efficient means for *consistent indexing* and *retrieval* of information [14].

Thesaurus terms are considered as the *"representation of concepts in the form of a noun or a noun phrase"* [20]. Concepts are perceived by thesaurus developers as referring collectively to a set of objects (*concept instances*) [30] that are considered as such not with respect to a formal classification process but through a common agreement. Under this perspective, the interpretation of a thesaurus term is a set of objects, which we will call the *extension* of the term. Thesauri are said to be structured since they include a fixed set of semantic term relationships. Due to the set theoretic definition of terms, these semantic relationships are interpreted as relations between sets [13, 21, 36].

The ISO 2788 Standard [20] for the documentation and establishment of monolingual thesauri defines the following four kinds of term relationships which distinguish structured thesauri from arbitrary collections of terms :

1. *generalization (broader term generic - btg)*,
2. *instance (broader term - bt)*,
3. *partitive* or *part-of (broader term partitive - btp)*,
4. *associative (related term - rt)* and
5. *equivalence (used for term - uf)*.

Term relationships *btg* and *btp* are called *hierarchical*. In this paper we are only concerned with *btg* relationships[2] which carry subset semantics and are the most frequently used hierarchical relationships. *Btg*-relationships are transitive and organize terms with similar semantics into directed acyclic graphs (DAG), referred to as *hierarchies*, or *classification schemes*. Two examples of *btg*-hierarchies are shown in Figures 4 and 5. For example in Figure 4, term *paintings* is a broader term of *oil paintings*, with the interpretation that all objects that belong to the extension of the latter, belong also to the extension of the former. A hierarchy is defined by its *root term*, a term with no broader term (*<visual works>* in Figure 4). We only assume mono-hierarchical thesauri, i.e. each term has exactly one broader term.

In the following, we will consider a *thesaurus* as a set of hierarchies, organized using the hierarchical *btg*-relationship. Although the definition we give is not complete w.r.t. all possible term relations existing in real thesauri it is sufficient for creating rich metadata schemata.

Definition 2. *A* thesaurus *is a couple* $\mathcal{T} = (D, btg)$ *such that*

[2] The interested reader can refer to ISO 2788 [20] for a deeper presentation of the remaining term relationships.

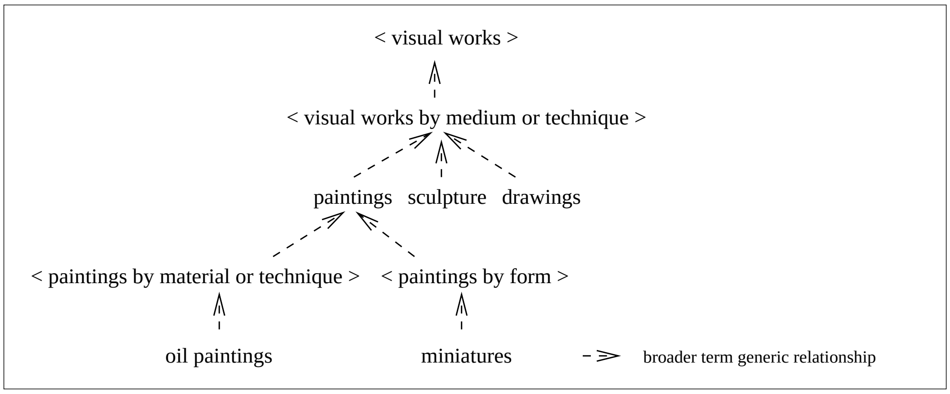

Fig. 4. Part of the Art & Architecture Thesaurus hierarchy *Visual Works* which collects all artifacts that are used for visual communication (paintings, sculptures, photos).

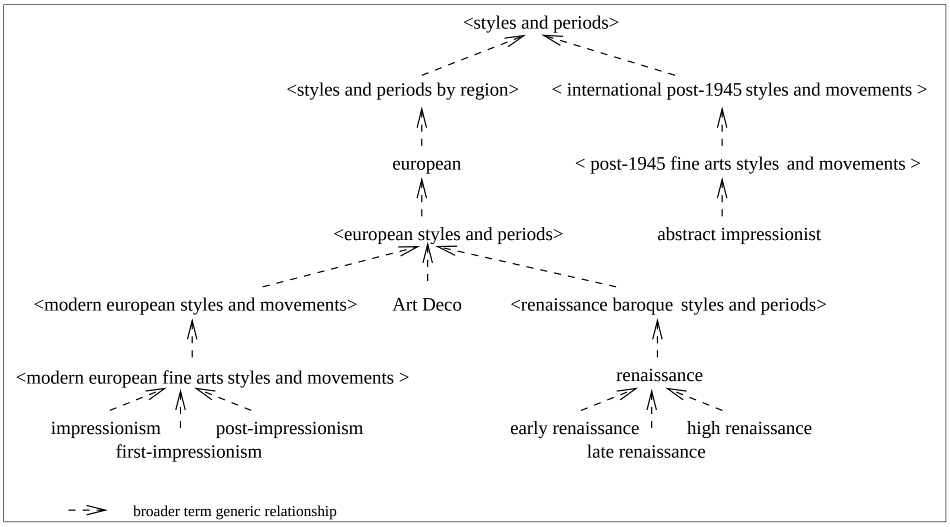

Fig. 5. Part of the Art & Architecture Thesaurus hierarchy *Styles & Periods* which collects all styles, periods and movements of Art in the western world.

1. $D = \{t_1, t_2, \ldots t_n\}$ *is a set of terms,*
2. btg is a binary relationship between terms such that for each pair of terms (t,t') there exists at most one btg-path between t,t' (mono-hierarchical thesaurus).

6 Creating RDF Schemas from Ontologies and Thesauri

In this section, we present a methodology for the construction of RDF schemas based on the integration of ontologies and thesaurus hierarchies. The construction of an RDF schema is done in three steps. In a first step, we specify for each ontology concept, a set of terms, the latter considered as *sub-concepts* of the former. This step is similar to establishing *inter-schema assertions* [8, 11] for database schema integration and cannot be a completely automated process since it requires the knowledge of the thesaurus and ontology semantics. Nevertheless, we must note that this knowledge could be partially derived from source data when thesauri are used to index concept instances. In a second, intermediate step, we extract *automatically* for each concept a *concept thesaurus*. This thesaurus contains only the terms connected to this concept by the connection relation, along with *broader-generic* relationships derived from the initial thesaurus. This process can be done automatically and does not require the knowledge of the ontology. In the final step we integrate these thesauri with the ontology to produce an RDF schema consisting of (1) a *structural view* provided by the ontology, (2) *connection relations* between concepts and terms, and (3) *thesaurus hierarchies*.

Observe that by the intermediate step it is possible to construct the resulting schema incrementally by extracting on demand concept thesauri that correspond to different ontology concepts. Moreover, another benefit of the separation of the integration into several steps is that we are able to monitor the result at any level of the integration process. Last, it is important to mention that our methodology is not related to a specific implementation platform.

6.1 Step 1 : Specialization of Concepts with Terms

In the first step of the integration process, thesaurus terms are “connected” to ontology concepts. These connections have inclusion semantics and are represented by a binary *connection relation* $Con \subseteq T \times C$ over a set of thesaurus terms T and a set of ontology concepts C. An example of a connection relation is presented in Figure 6. Terms *impressionism*, *post-impressionism* and *abstract impressionism* of the *Art & Architecture Thesaurus* hierarchy *Styles & Periods* (Figure 5) describe specific styles (ontology concept `Style` in Figure 3). Term *first-impressionism* of the same hierarchy describes both a style and a period (concepts `Style` and `Period` respectively). Similarly, term *renaissance* and its narrower terms of *Styles & Periods* hierarchy describe different types of styles and periods (ontology concepts `Style` and `Period` respectively). Finally, terms *paintings*, *oil paintings* and *sculpture* of the AAT hierarchy *Visual Works* (Figure 4) define different kinds of iconographic objects (ontology concept `Iconographic Object`).

Term	`Concept`	*Term*	`Concept`
impressionism	`Style`	*paintings*	`Iconographic Object`
post-impressionism	`Style`	*oil paintings*	`Iconographic Object`
abstract impressionism	`Style`	*sculpture*	`Iconographic Object`
renaissance	`Style`	*early renaissance*	`Style`
renaissance	`Period`	*early renaissance*	`Period`
late renaissance	`Style`	*high renaissance*	`Style`
late renaissance	`Period`	*high renaissance*	`Period`
first-impressionism	`Period`	*first-impressionism*	`Style`

Fig. 6. A connection relation *Con* for AAT hierarchies *Styles & Periods*, *Visual Works* and ontology concepts `Style`, `Period` and `Iconographic Object`.

In the following we will say that, if term t is connected to concept c, it is labeled by c. The way the user actually labels terms (chooses concepts to be connected to a given term) will not be discussed in this paper because of lack of space. Briefly speaking, either one labels t with some concept c with the assumption that all descendants of t are connected to (labeled with) c or one chooses explicitly among the descendants of t.

In the previous example, we do not connect the whole thesaurus hierarchy *Styles & Periods* to concepts `Style` and `Period`. We adopt this *selective approach*, i.e. relating thesaurus terms to ontology concepts explicitly, for several reasons. An obvious reason is that some terms could be out of the scope of the application that has to be described by the resulting RDF schema. For example, if some application is only concerned with paintings, then terms referring to artifacts other than paintings (e.g. *sculpture*, *drawings*) need not be considered in the resulting schema. Another reason is that some terms (e.g. *guide terms* in [20, 38]) are used to organize thesaurus hierarchies (e.g. *<visual works by medium or technique>*) and might have no use for describing information. Finally, another important reason is that thesaurus hierarchies might contain terms which can be connected to different concepts. For example, terms of the AAT hierarchy *Styles & Periods* (Figure 5) describe styles (e.g. *impressionism*), periods (e.g. *art deco*), or both styles and periods (e.g. *renaissance*). Connecting terms to concepts in a selective manner allows users to clarify between the multiple semantics of a term (e.g. as in the case of *homonyms*) and consequently resolve semantic ambiguities at the thesaurus level.

6.2 Step 2 : Thesaurus Extraction

After having defined the connection relations between terms and concepts, we extract for each concept in the connection relation, a thesaurus, called *concept* *thesaurus*. This is done in two steps. First, each term in thesaurus $\mathcal{T}$ is labeled by the concepts to which it is connected in the relation Con. Observe that a term can be connected to several concepts, i.e. term labels are sets of concept names. For example, in the connection relation illustrated in Figure 6, term *first-impressionism* is connected to both `Style` and `Period` concepts. In this case, the label of term *first-impressionism* is the set of concepts {`Style, Period`}.

Second, we define a selection operation σ that constructs from a labeled thesaurus $\mathcal{T}_\lambda$ and a set of concept names S a new labeled thesaurus that contains (1) the set of terms in $\mathcal{T}_\lambda$ whose labels contain at least one concept in S and (2) *btg* relations between these terms, induced by the *btg* relations in the initial thesaurus. More precisely :

Definition 3. *Let $\mathcal{T}_\lambda = (D, btg)$ be a labeled thesaurus where each term is labeled by a (possibly empty) set of concepts : $\lambda\ : D \rightarrow 2^C$. Let S be a set of concept names. The selection $\sigma(S, \mathcal{T}_\lambda)$ creates a new thesaurus as follows :*

1. *keep all terms t of $\mathcal{T}_\lambda$ which are labeled by at least one concept name in S ($S \cap \lambda(t) \neq \emptyset$),*
2. *create btg relations between all terms t and t' in $\sigma(S, \mathcal{T}_\lambda)$ which are related by a btg path in $\mathcal{T}_\lambda$ that contains no term in $\sigma(S, \mathcal{T}_\lambda)$.*

A naive algorithm calculating $\sigma(S, \mathcal{T}_\lambda)$ is shown in Appendix A. Since $\sigma(S, \mathcal{T}_\lambda)$ can be evaluated without the knowledge of the underlying ontology, this algorithm can be executed on the thesaurus site without accessing information from the ontology. This property is useful in a distributed environment where thesauri and ontologies might be stored on different sites.

Using this selection operation it is possible to define a labeled thesaurus $\mathcal{T}_\lambda^c$ for each concept c in the set of ontology concepts C as follows :

Definition 4. *Let $\mathcal{T}_\lambda = (D, btg)$ be a labeled thesaurus. Let c be a concept and S_c be the set of sub-concepts of c in C including c. Then, we can define a labeled* concept thesaurus *$\mathcal{T}_\lambda^c = \sigma(S_c, \mathcal{T}_\lambda)$ which contains all terms connected (in Con) to c or a sub-concept of c.*

For the definition of a concept thesaurus we exploit not only the *btg* relations between the terms but also the isa relationships at the ontology level. Consider the example in Figure 7. Term v is labeled by concept d, term t by c and w by e. The selection operation on concept c will construct the thesaurus $\mathcal{T}_\lambda^c$ that contains besides term t, terms v and w that are labeled by its sub-concepts. Observe also that a term can appear in multiple concept thesauri, and terms that are not labeled by any concept have disappeared from the concept thesauri. For example, term u is not connected to any concept and has disappeared from the concept thesauri in Figure 7. Moreover, the selection operation on concept c created a *btg* relation between terms w and t which were not directly related in the original thesaurus.

Each concept thesaurus can be extracted independently and contains only a subset of the terms defined in the connection relation. This means that the

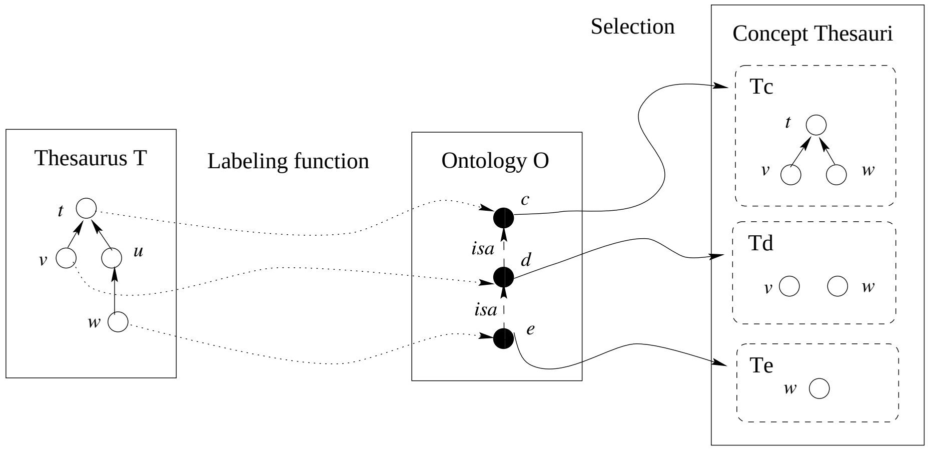

Fig. 7. Extracted Thesaurus Examples.

size of the extracted thesauri is bound by the number of terms in the connection relation and is independent of the size of the original thesaurus.

At this point, we should mention that a concept thesaurus can be induced by those of its super-concepts. For example, if d is a sub-concept of c, and $\mathcal{T}_\lambda^c$ is the concept thesaurus of c, then the concept thesaurus of d can be extracted as follows : $\mathcal{T}_\lambda^d = \sigma(S_d, \mathcal{T}_\lambda^c)$. In the previous example, only concept thesaurus $\mathcal{T}_\lambda^c$ of concept c has to be extracted from the original thesaurus $\mathcal{T}_\lambda$. All thesauri corresponding to sub-concepts of c might then be created on demand during the creation of the RDF schema (Section 6.3).

6.3 Creation of the RDF Schema

In this section we will present how the RDF schema is constructed out of a set of *concept thesauri*. This schema will incorporate the set of *ontology concepts* and *roles*, the *concept thesauri* defined for each ontology concept and *connections* between terms and concepts. In short, ontology concepts and thesaurus terms are modeled as RDF *classes*, ontology roles as RDF *properties*. Ontology *isa* relationships, *connection relations* between terms and concepts and *btg* relations between terms all carry inclusion semantics and are modeled with the RDF *subclassOf* property.

The creation of the RDF schema $\mathcal{S}$ for an ontology $\mathcal{O} = (C, R, isa)$, and a set of concept thesauri $\mathcal{T}_\lambda^c = (D, btg)$ is straightforward :

1. The set of RDF *classes* in $\mathcal{S}$ is obtained as follows:
   (a) for each ontology concept c define RDF class `c`,
   (b) for each term t in $\mathcal{T}_\lambda^c$ define RDF class `c:t`,

2. The set of RDF properties is obtained by defining for each typed role $r(c, d)$ in R an RDF property with `domain` RDF class `c` and `range` RDF class `d`.
3. The set of RDF `subclassOf` properties is obtained as follows:
   (a) for each $isa(c, d)$ relationship between ontology concepts c and d define an RDF `subclassOf` property between RDF classes `c` and `d`.
   (b) for each RDF class `c:t`, corresponding to a root term in thesaurus $\mathcal{T}_\lambda^c$, add an RDF `subclassOf` property between RDF classes `c:t` and `c`.
   (c) for each *btg* relation between two terms t and t' in a concept thesaurus $\mathcal{T}_\lambda^c$, define an RDF `subclassOf` property between RDF classes `c:t` and `c:t'`.

It is interesting to note that we connect only the *root* term of each concept thesaurus to the corresponding concept. Due to the transitivity of the RDF *subclassOf* property, it can be induced that a term t is a *subclassOf* another term t or a concept c.

The RDF schema illustrated in Figure 8 has been constructed from the ontology in Figure 3, the thesaurus classification schemes in Figures 4, 5, and the connection relation in Figure 6.
Ontology concepts `Man Made Object`, `Iconographic Object`, `Style`, `Period` and terms *oil paintings*, *paintings*, *impressionism* and *first-impressionism* are all represented as RDF classes (lines `5,7,9,11,21,23`). For simplification, we only prefix terms with the corresponding concept if they are contained in different concept thesauri. RDF class `paintings` is defined as a subclass of class `Iconographic Object` (line 22), since term *paintings* is the root term of `Iconographic Object` concept thesaurus. In the same way, classes `impressionism` and `first-impressionism` are defined as subclasses of concepts `Style` and `Period` respectively (lines `26,28`). Class `oil paintings` is a subclass of class `paintings` (line 24) (defined by the *btg*-relations between term *oil paintings* and term *paintings*). Ontology role *style*, is defined as an RDF property, its domain being the class `Iconographic Object` (line 19) and its range class `Style` (line 20). By definition of the *subclassOf* property, all subclasses of `Iconographic Object` inherit this property.

Using this RDF schema, one can provide RDF descriptions about specific web resources. For example, a new RDF description for the source described in Figure 1 is shown in Figure 9. When comparing this new description with the previous one, one can observe that we have replaced namespace `artifact` by a new namespace `int` which corresponds to the RDF schema in Figure 8. In this RDF description, semantic information that was captured as a value in the previous description has been added at the schema level. For example, the fact that the resource described an impressionist painting was encoded in the value of tag `<artifact:style>`. This value corresponds in fact to a term in the AAT and is represented as an instance of class `int:impressionism` (line `11`) in the new schema. The same argument holds for the value *first-impressionism* which is now represented as an instance of RDF class `int:first-impressionism` (line `12`). Observe also (tag `<rdf:Description>`) (Figure 1, line 7), has been replaced by a typed node tag `<int:oil paintings>` (line 8) indicating that the described resource is an oil painting.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
         xmlns:int="">
  <rdfs:Class rdf:ID="Physical Object"></rdfs:Class>
  <rdfs:Class rdf:ID="Man-Made Object">
    <rdfs:subclassOf rdf:resource="#Physical Object"/></rdfs:Class>
  <rdfs:Class rdf:ID="Iconographic Object">
    <rdfs:subclassOf rdf:resource="#Man-Made Object"/></rdfs:Class>
  <rdfs:Class rdf:ID="Period"></rdfs:Class>
  <rdfs:Class rdf:ID="Title"></rdfs:Class>
  <rdfs:Class rdf:ID="Style"></rdfs:Class>
  <rdf:Property rdf:ID="of-period">
    <rdfs:domain rdf:resource="#Man Made Object"/>
    <rdfs:range  rdf:resource="#Period"/></rdf:Property>
  <rdf:Property rdf:ID="title">
    <rdfs:domain rdf:resource="#Man Made Object"/>
    <rdfs:range  rdf:resource="#Title"/></rdf:Property>
  <rdf:Property rdf:ID="style">
    <rdfs:domain rdf:resource="#Iconographic Object"/>
    <rdfs:range  rdf:resource="#Style"/></rdf:Property>
  <rdfs:Class rdf:ID="paintings">
    <rdfs:subclassOf rdf:resource="#Iconographic Object"/></rdfs:Class>
  <rdfs:Class rdf:ID="oil paintings">
    <rdfs:subclassOf rdf:resource="#paintings"/></rdfs:Class>
  <rdfs:Class rdf:ID="impressionism">
    <rdfs:subclassOf rdf:resource="#Style"/></rdfs:Class>
  <rdfs:Class rdf:ID="first-impressionism">
    <rdfs:subclassOf rdf:resource="#Period"/></rdfs:Class>
</rdf:RDF>
```

Fig. 8. The RDF schema resulting from the integration of the ontology and thesaurus.

```
<rdf:RDF
 xmlns:web-page ="http://metalab.unc.edu/louvre/namespaces/web-pages"
 xmlns:int ="http://www.connectit.com/icom/aat">
 <rdf:Description
  about="http://metalab.unc.edu/louvre/paint/monet/first/highway/">
  <web-page:title>Web Museum: Monet, Claude :Impression :
                                        soleil levant</web-page:title>
   <web-page:presents>
    <int:oil paintings
     about="http://metalab.unc.edu/louvre/paintings/monet/impression">
     <int:title>Impression : soleil levant</int:title>
      <int:style><int:impressionism/></int:style>
      <int:of-period><int:first-impressionism/>
      </int:of-period>
   </int:oil paintings>
   </web-page:presents>
   <web-page:creator>Nicolas Pioch</web-page:creator>
  </rdf:Description>
</rdf:RDF>
```

Fig. 9. RDF description for Claude Monet painting using the integrated schema.

7 Conclusions and Future Work

In this paper, we have presented a modular, component based approach to the construction of RDF schemas based on the integration of ontologies and thesaurus hierarchies. Our examples were taken from the cultural application domain, however the presented approach can also be applied to other semantically rich scientific (e.g. medicine, biology or chemistry) or electronic commerce (e.g. electronic catalogue) applications.

An interesting issue concerns the specification of the connection relation. Whereas, this relation can be specified manually for a limited number of terms and concepts, its creation gets cumbersome when the number of connected terms and concepts increases. There are two possible solutions to this problem. First, as already mentioned, thesaurus terms are used for indexing documents and other actual data. The existence of these terms at the data level might then be exploited for the automatic creation of the connection relation by using data mining techniques. The second solution consists in the definition of a query language for thesauri which allows to extract sets of terms by simple declarative queries (e.g. path expressions). In this way, the connection relation can be represented as a conceptual view on the thesaurus where concepts are defined by term queries.

The resulting RDF schema can be perceived as the domain model in mediation based systems such as Information Manifold and SIMS and it plays an essential role in achieving *semantic interoperability* between the sources. This domain model provides a uniform view of information in the domain of discourse.

Users pose queries against this model and information sources export their content descriptions as views on this model. The source descriptions, are used by mediators to identify the set of relevant information sources with respect to a user query. We intend to apply our approach to the definition of semantically rich domain models in the context of the Artemis project [3], which proposes a flexible framework for the definition of cultural mediators. In the same application context, we consider that an important issue is the exploitation of the *related term* (synonym) relation for query rewriting and concept subsumption. In this direction, we intend to study more thoroughly the integration of *multilingual thesauri* and the exploitation of inter-thesaurus relations for creating multi-lingual RDF schemas. Such schemas can be used effectively in the context of querying multi-lingual information sources.

A first prototype is currently under implementation. We have already implemented a loader of RDF descriptions into the O_2 object-oriented database management system of Ardent Software using the SiRPAC [37] RDF parser to parse RDF documents (RDF schemas and descriptions). The next step is to provide tools for specifying the connection relation and create concept thesauri. Finally, we want to provide a query interface based on OQL for selecting resources according to their descriptions.

Acknowledgments We would like to thank Michel Scholl and Vassilis Christophides for their precious comments on preliminary versions of this paper. The authors also thank Serge Abiteboul, Marie-Christine Rousset, Rodney Topor, and Anne-Marie Vercoustre for fruitful discussions. Last, but not least we would like to thank Martin Doerr for his help in the understanding of the ICOM/CIDOC Reference Model.

References

1. The Art & Architecture Thesaurus. http://www.ahip.getty.edu/vocabulary/-aat_intro.html.
2. ACM Computing Classification System. http://www.iicm.edu:8080/jucs_classification.
3. B. Amann, V. Christophides, I. Fundulaki, M. Scholl, and A.M. Vercoustre. Intelligent Mediation of Cultural Information Sources. *ERCIM News*, (35), October 1998.
4. J. Ordille A.Y. Levy, A. Rajaraman. Querying heterogeneous information sources using source descriptions. pages 251–262, Bombay, India, September 1996. Morgan Kaufmann.
5. C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323–364, December 1986.
6. T. Bray. RDF and Metadata, June 1998. http://www.xml.com/xml/pub/98/06/-rdf.html.
7. D. Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification. Technical Report REC-rdf-model-19990303, W3C, March 1999. W3C Proposed Recommendation.

8. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Description logic framework for information integration. To appear in Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98).
9. C. Y. Chee, Y. Arens, C. A. Knoblock, and C. N. Hsu. Retrieving and Integrating Data from Multiple Information Sources. *International Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.
10. Les dossiers de l'inventaire général. http://aquarelle.inria.fr/Inventaire.
11. C. Collet, M. Huhns, and W. Shem. Resource Integration Using a Large Knowledge Base in Carnot. *IEEE Computer*, pages 55–62, December 1991.
12. Dublin Core Metadata Initiative. http://purl.oclc.org/dc/.
13. M. Doerr and I. Fundulaki. A proposal on extended interthesaurus links semantics. Technical Report TR-215, Institute of Computer Science-FORTH, March 1998.
14. D.J. Foskett. *Readings in Information Retrieval*, chapter Thesaurus. Morgan Kaufmann, 1997.
15. Thomas. R. Gruber. A Translation Approach to Portable Ontology Specifications. Technical Report KSL 92-71, Knowledge Systems Laboratory, Computer Science Department, Stanford University, April 1993.
16. N. Guarino. Understanding, Building, and Using Ontologies. A commentary to "Using Explicit Ontologies in KBS Developemtn", by Heijst, Schreiber, and Wielinga.
17. N. Guarino. *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, chapter Semantic Matching : Formal Ontological Distinctions for Information Organization, Extraction, and Integration, pages 139–170. Springer Verlag, 1998.
18. N. Guarino, C. Masolo, and G. Vetere. OntoSeek: Using Large Linguistic Ontologies for Gathering Information Resources from the Web. Techical report, LADSEB, March 1998. Submitted for Publication.
19. International Guidelines for Museum Object Information: The CIDOC Information Categories. http://www.cidoc.icom.org/guide/.
20. Documentation - Guidelines for the establishment and development of monolingual thesauri. International Organization for Standardization, 11 1986. Ref. No ISO 2788-1986.
21. Documentation - Guidelines for the establishment and development of multilingual thesauri. International Organization for Standardization, 2 1985. Ref. No. ISO 5964-1985.
22. K. Jeffery. Metadata: An Overview and some Issues. *ERCIM NEWS*, (35), October 1998.
23. Knowledge Interchange Format (KIF). http://logic.stanford.edu/kif/kif.html.
24. K. Knight and S. Luk. Building a Large-Scale Knowedge Base for Machine Translation. In *Proc. of the American Association of Artificial Intelligence AAAI-94*, Seattle, WA, 1994.
25. C. Lagoze. The Warwick Framework : A Container Architecture for Diverse Sets of Metadata. *D-Lib Magazine*, July/August 1996.
26. O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. Technical Report REC-rdf-syntax-19990202, W3C, February 1999. W3C Proposed Recommendation.
27. The Library of Congress Subject Headings. http://www.grci.com/services/-library/libcongress/index.shtml.
28. D. B. Lenat. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

29. P. Martin. The WebKB set of tools. Technical report, Griffith University, School of Information Technology, Australia, October 1997. available at: http://meganesia.int.gu.edu.au/ phmartin/WebKB/doc/index.html.
30. R.S. Michalski. *Categories and Concepts, Theoretical Views and Inductive Data Analysis*, chapter Beyond Prototypes and Frames: The Two-Tiered Concept Representation. Academic Press, 1993.
31. A. Michard, V. Christophides, M. Scholl, M. Stapleton, D. Sutcliffe, and A-M. Vercoustre. The Aquarelle Resource Discovery System. *Journal of Computer Networks and ISDN Systems*, 30(13):1185–1200, August 1998.
32. G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
33. R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. Enabling Technology for Knowledge Sharing. *AI Magazine*, Fall 1991.
34. W3C Technology and Society Domain : Resource Description Framework (RDF). http://www.w3.org/RDF/.
35. A. Sheth. *Interoperating Geographic Information Systems*, chapter Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. Kluwer, 1999.
36. M. Sintichakis and P. Constantopoulos. A Method for Monolingual Thesauri Merging. In *Proc. 20th International Conference on Research and Development in Information Retrieval, ACM SIGIR*, Philadeplphia PA, USA, July 1997.
37. SiRPAC - Simple RDF Parser & Compiler. http://www.w3.org/RDF/-Implementations/SiRPAC/.
38. D. Soergel. The Art and Architecture Thesaurus, AAT. A critical appraisal. Technical report, College of Library and Information Sciences, University of Maryland, 1995.
39. Unified Medical Language System. http://gmedserv.nlm.nih.gov/research/umls/.
40. MARC STANDARDS. http://lcweb.loc.gov/marc/marc.html.
41. Stuart Weibel. Metadata: The Foundations of Resource Description. *D-Lib Magazine*, July 1995.
42. W3C Technology and Society Domain : Extensible Markup Language (XML). http://www.w3.org/XML/.

A A Naive Algorithm for Thesaurus Extraction

Input : (i1) set of concepts S
(i2) a thesaurus $\mathcal{T} = (T, btg)$
(i3) a labeling function λ
(i4) a set of root terms R in $\mathcal{T}$
Output : (o1) a thesaurus $\sigma(\mathcal{T}_\lambda)$
Procedure :

```
$T_S := \emptyset$; $btg_S := \emptyset$
for all terms r in R
    for all narrow terms t of r
        create_btg(r, t)

create_btg(r, t) : if λ(r) ∩ S = ∅
    u = t
else
    add r to T_S
    if λ(t) ∩ S ≠ ∅
        add t to T_S
        add (t, r) to btg_S
        u = t
    else
        u = r
for all narrow terms t' of t
    create_btg(u, t')
```