A Metadata Approach to Manage and Organize Electronic Documents and Collections on the Web

Ana Maria de Carvalho Moura
Genelice da Costa Pereira
Instituto Militar de Engenharia - IME/RJ
Departamento de Engenharia de Sistemas
[anamoura,genelice]@ime.eb.br

Maria Luiza Machado Campos
Universidade Federal do Rio de Janeiro - UFRJ
Departamento de Ciência da Computação
mluiza@nce.ufrj.br

Abstract *In recent years, the number of information sources offered on the Web has grown tremedously. Support for accessing these information sources has mostly been concentrated on browsing and search tools. Digital libraries and Web directories constitute important initiatives to improve information access, creating and organizing document collections hierarchically, according to different criteria. Search tools, on the other hand, offer a more comprehensive coverage of resources, using robot-based services to collect and index documents, that can be latter accessed using information retrieval techniques. However, technologies applied to search mechanisms on the Web still offer little support to manage document collections, as the association between these documents cannot be explicitly identified, neither by their formats nor by their types. This paper presents a formal structure for organizing and describing collections and their documents on the Web. It is based on a metadata conceptual model which explores relationships between information resources at different levels of granularity. To validate this model, a prototype has been implemented using both a semi-structured and an object-relational database (DB) approach.*

Keywords: *metadata, digital library, RDF, XML, electronic documents, information retrieval*

1 Introduction

The Web has become a major source of information in all areas of interest. However, users with different levels of expertise began to disseminate vast amounts of diversified types of resources, leading to an information overhead. Automatic indexing strategies of document contents used by most of current search tools do not answer users' expectations. It is not rare for a user to retrieve useless information, while other relevant resources are not indexed at all. The effectiveness of these indexing structures depends highly on the way resources are described by their providers.

There already exists a major agreement of the research community that the use of metadata is the adequate solution to promote more efficient and accurate retrieval services on the Web, making it possible the integration and information exchange amongst heterogeneous digital sources. In order to provide better management of resources on the Web, many metadata standards have been created and adapted to answer users' needs on describing specific resources (MARC, EAD, TEI, GILS, SOIF, Dublin Core (DC), IAFA, etc.). These standards comprise not only bibliographic data such as location, author, format, publication date but also more detailed and elaborated descriptive data, which can include metadata for resource administration and control,

availability and usage. Moura et al. in [17] presents a comprehensive survey on the subject.

Due to the diversity of existing resources on the Web there is already a consensus that it is not possible to adopt a unique metadata standard to describe all these kind of resources, as it will not be able to include a comprehensive set of descriptors adequate to cover all the resources application domains.

Recent initiatives recognize the need for a higher-level container architecture that can accommodate different metadata standards already in use, establishing general frameworks where these standards could coexist. The Warwick Framework [15], and the Meta Content Framework [8] are examples of such initiatives. They accommodate data and metadata in packages, interrelating data, descriptors and schemes of description on a flexible architecture. More recently, RDF (Resource Description Framework) [22] has been proposed to provide unambiguous methods of expressing semantics and as a means for publishing both human-readable and machine-processable vocabularies among information communities. RDF uses a high level meta-model that does not impose semantics to any resource description community, but rather provides the ability for these communities to define metadata elements as needed. RDF uses XML (eXtensible Markup Language) as a common syntax for the exchange and processing of metadata [22].

This paper presents a metadata model to describe and organize electronic documents and collections on the Web, which has been validated by the development of a document management system. This system has been implemented using both a traditional database environment as well as an XML/RDF approach.

The rest of this paper is organized as follows. Section 2 presents a brief overview of current technologies used to search and organize resources on the Web. Section 3 describes a hierarchical structure to organize resources on the Web, whose main concepts have been used to create a metadata model to manage documents and collections in this environment. Section 4 shows an example exploring the constructs of this model. Section 5 describes the prototype development. Finally, section 6 concludes with additional comments and future work.

2 Organizing, Searching and Retrieving Resources on the Web

The Web is actually considered a major source of information in all domains. Due to its heterogeneous and distributed nature, a huge amount of documents and collections of all sorts (databases, programs, papers in different formats, personal mails, search results, multimedia, gopher and ftp files, etc.) is made available in an autonomous way.

Search mechanisms to enhance the quality of information retrieval are considered as an important challenge for the scientific community and a fundamental tool for Web users. The effectiveness of these tools depends directly on the way resources have been cataloged on the Web. Documents can be organized using Web directories, databases or other digital libraries techniques, whose contents can be handled using different retrieval mechanisms to provide better services to the users. Search tools are classified into four categories [9, 11]:

(i) **Directories**: search for information by subject matter in a hierarchical search which starts from a general subject heading and follows with a succession of more specific sub-headings. Encyclopedia Britannica, Yahoo, Cadê are good examples of this category;

(ii) **Search engines**: search for information through use of keywords, giving a list of references or hits as results. Their scope is substantially larger than that of a directory search tool. Keyword searches require more explanation than subject searches, because of their broader scope and greater complexity. Alta Vista, Google, Lycos, Infoseek, among many others, are examples of search engines;

(iii) **Directories with search engines**: use both search and keyword search methods interactively. In the directory search part, search follows as described in directories, but at each stop along the hierarchical search, the option to use a search engine is provided to enable the searcher to convert to a keyword search. By narrowing the search field, it is possible to have more relevant results. Yahoo[1], Magellan are examples of tools in this category;

(iv) **Multi-engine search tools (meta-searchers)**: use a number of search engines in parallel. The search is conducted via keywords, using common operators or plain language. Results are integrated in a single list, providing fewer hits of likely greater relevance. Metacrawler and Dogpile are classified in this category.

[1] Yahoo is classified in both (a) and (c) categories.

Digital Libraries are usually built using some of these tools and constitute a friendly environment for users who need information on specific knowledge domains. They have achieved great importance in recent years as an adequate framework to organize documents and collections on the Web, as presented next.

2.1 Use of Digital Libraries to Organize Documents on the Web

Digital Libraries (DL) are organized collections of information stored in digital media whose structure resembles a traditional library, providing a large set of materials and services. In recent years, the use of DL has tremendously increased as an advantageous alternative to specialized sites on the Web. One important reason is the availability of services such as electronic catalogs, built by professionals from different domains, whose access may be considered similar to traditional libraries. In this sense, DLs enhance some of the deficiencies found on search tools (inadequate results out of the expected context, for example), providing a full control and management of resources available in their collections [24].

In this context, metadata play an important role on DL management. They provide support for identification, description and location of network electronic resources, whose characteristics are not effectively supported by current search mechanisms. Metadata should be produced and associated to Internet resources so that metadata-aware search services could be developed. In reality, a mix of full text indexing and metadata-based content retrieval could be used to provide better and more precise retrieval results.

In order to take advantage of the organization and catalog services developed for DL management, some very important DL projects have been investigated [24], such as those which are part of the DLI Program [6], a consortium of universities, public institutions, and private corporations, like Alexandria, Berkley, Michigan, Illinois, Stanford, Carnegie Mellon, as well as other initiatives such as the American Congress Library [14] and NCSTRL (Networked Computer Science Technical Reference Library) [19]. Each one of these DL emphasizes a specific knowledge domain. They all use metadata to catalog their materials and to enhance the quality of information retrieval, basically constituted of direct search and navigation.

3 Organization Structure for Electronic Documents and Collections on the Web

As commented previously, technologies such as WWW, Gopher and anonymous FTP offer little structure to manage document collections, as most of these technologies are based on the file hierarchy abstraction. If different files compose a unique document, the association between these files cannot be explicitly identified, neither by their formats nor by their types, due to a lack of standardization. According to Lagoze [13] a possible solution to enable control at the intellectual content level would be to use an abstract structure, hiding from the users any file structural details. Coherent organizational structures imposed on description are necessary to provide a view that supports navigation.

We propose a metadata model for describing electronic documents as an extension of the work developed by Barreto [1], where six important aspects are considered: *structure* (a document may be composed of many parts, and visualized according to different levels of granularity); *intellectual content* (expresses the document subject); *relationships* (a document may be related to several other documents or resource types); *internal organization* (the same intellectual content may have different forms of expression); *external organization* (a document may be organized into different collections, depending on its subject matter); and *presentation formats* (such as HTML, or PostScript). Furthermore, according to important metadata architectures (Warwick, MCF, for example) collections and electronic documents in the proposed model are considered as digital objects.

Figure 1 presents the abstract levels of a digital object according to a multi-level hierarchical structure. This structure helps to visualize how electronic resources can be made available on the Web, represented at the top level of this hierarchy. It also takes into account two broad relationship categories: **structural**, representing associations among components of a document or collection; and **contextual**, representing document or collection relationships with other complementary resources (bibliographical references, descriptions, distinct versions of the document intellectual content, copies of the original document, etc.).

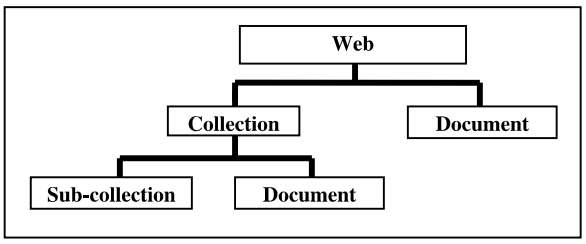


Figure 1: Abstract levels of a digital object according to a multi-level hierarchical structure

According to Tillet's analysis [25], intellectual content expresses the highest abstraction level to characterize a document or a collection. A collection expresses its external organization. It is an aggregation of electronic items organized according to the user's static vision (author, subject, title, etc.). At the collection/sub-collection layer documents may be organized according to standard classification schemas. Metadata to describe collections may comprise, for example: its subject, search methods, etc. Contextual relationships, representing documents association with other complementing resources, may include for instance, notes, bibliographic references, etc.

Documents can be independent of a collection. It can be visualized according to three aspects. First, **document typology**, which determines its internal organization, specifying its type or content expression (paper, thesis, technical report, etc.). Contextual relationships at this level comprise, for example, terms and conditions for content access. The second aspect, **format**, specifies the different forms the same document is made available (doc, pdf, ps, jpeg, etc.). The formats related to a unique document characterize the *physical personifications* used to disseminate its content [25]. Contextual relationships may include, for example, translations to other languages or document copies available in different providers. Finally the third aspect, **structure**, represents the structural views on which a physical personification may be segmented for presentation and research. These views can be: *physical* corresponding to document division into physical parts, such as pages, frames, blocks or physical coordinates; *logical*, consisting of a hierarchy of document segments, each of them corresponding to a distinct semantic component, such as a header, a paragraph, a section, etc.; and *hypermedia*, describing the semantic nature of hyperlink associations of a network document. Consider, for example, a document in HTML format composed of a text, an image and several links to external references.

Among the metadata standards investigated in the scope of this work, IAFA, DC and RFC 1807 [24] have been chosen as the most adequate to describe documents on the Web. Together they comprise a meaningful set of descriptors distributed in the layers described above. Figure 2 shows this classification, an important feature for the metadata model described next.

Metadata Standards	Layers			
	Collection	Content Expression	Physical Personification	Structure
IAFA	Template-Version, Template-Type, Handle, Title, Short-Title, Author-(USER*), Size-v*, URI-v*, Requirements, Creation-Date, Source, Format-v*,Admin-(USER*), Publisher-(Organization*), Library-Catalog-v*, Description, Keywords, Last-Revision-Date, Language-v*	Template-Version,Template-Type,Description, Handle, Title, Short-Title, Author-(USER*), Creation-Date, Admin-(USER*), Source, Size-v*, Requirements, Category, Bibliography, URI-v*, Copyright, Citation, Discussion, Version-v*, Keywords, Character-Set-v*, ISBN-v*, ISSN-v*,Last-Revision-Date-v*, Library-Catalog-v*, Last-Revision-Date, Publisher-(Organization*)*	Format-v*, Language-v*, URI-v*	
DUBLIN CORE	Title, Creator, Format, Contributor, Publisher, Date, Source, Type, Description, Subject, Language, Coverage, Identifier	Date, Publisher, Type, Title, Description, Creator, Contributor, Identifier, Relation, Source, Rights, Coverage, Subject	Language, Format	
RFC 1807	Id, Abstract, Date, Title, Handle, Entry, Keyword, End, Other_access, Language, Organization	Organization, Date, Revision, Type, Bib-version, Id, Abstract, Entry, Title, Author, Corp-Author, Contact, Copyright, Handle, Retrieval, Grant, Keyword, Period, Cr-category, Series, Funding, Monitoring, End, Contract, Notes, Withdraw, Other_access	Language	Pages

Figure 2: Metadata standard descriptors distributed into a multi-level hierarchical structure

3.1 MODDEC: the Proposed Metadata Model

This model suggests the use of an abstract structure whose physical structure details are completely hidden from users, whereas providing the complete control of resources based on their intellectual content. It takes into account the use of digital objects as a relevant point for managing resources. A resource is an information unity which recursively describes itself through the use of descriptor elements of metadata standards. A digital object works as a data and metadata container as in the Warwick architecture [15], providing modularity, distribution and recursivity. Furthermore, it is organized according to a hierarchical recursive representation constituted of documents, collections, metadata standards and associations between digital objects and their elements descriptors. Figure 3 presents the MODDEC schema represented in UML notation, whose components (classes) are briefly described next [20]. The description of some properties is omitted as they are self-explicative.

- **DigitalObject**: represents the superclass including all the resources covered by the model: documents, collections, and associations between these objects. The attribute *metadataContainer* describes contextual relationships between information resources. It allows the user to have access to the set of metadata descriptors of an object according to its type and corresponding level in the organization structure seen in Figure 1. Hence, each contextual relationship corresponds to an object of AssociationObject type. The attribute *dataContainer* describes structural relationships between information resources. It references digital objects in different levels of granularity according to the organization structure (Figure 1) or a unique url of the object in question. Notice that all the attributes of this class are inherited by its subclasses, providing metadata modularization. From this object it is possible to navigate hierarchically through all the structure components: collections and sub-collections, and documents according to their formats, typologies, etc., as well as having access to all objects related to them (references, copies, etc.);

- **DocumentObject**: determines how a document may be visualized according to its *content expression, physical personification* or *structure*. For each document view a new digital object is created. The attributes of each of the following subclasses are specified according to the metadata standard adopted for the description of that subclass, at its corresponding level:

  - **ExpressionObject**: represents the way a document is expressed (a paper, a book, a manual, a Master thesis, etc.), without considering any format or representation aspect. The attribute *metadataContainer* in this class may contain, for example, terms and conditions for access, versions etc. This attribute makes it possible a user to retrieve all the references of a book or the papers of a journal, for example. Attribute *dataContainer* enables users to get all the available formats of a paper or all the chapters of a thesis, for example;

  - **PersonificationObject**: specifies the different formats a document can be materialized (pdf, html, ps, etc.). An ExpressionObject generates at least a PersonificationObject. From the information obtained in the attribute metadataContainer the user can obtain, for example, the properties of a journal or its copy in another language. Attribute dataContainer provides queries such as "Get all the chapters of a thesis in PDF format" or "Get the contents of page 43 of a technical report in doc format". If the document does not have structural relationships, the object URL can be included in order to provide its direct access;

  - **StructuralObject**: determines the types of structural views a document can have at the structural level, considered as logical, physical and hypermedia. Again, if the document does not have structural relationships, the object URL can be included in attribute *dataContainer* in order to provide its direct access.

    - **LogicalObject:** specifies the logical structural unit such as a section or a paragraph, described by attribute *logicalPart*. Attribute *metadataContainer* makes it possible to search, for example, the copies of chapter X of a paper. Attribute *dataContainer* provides queries such as: "get all the pages of chapter 4 of a book", or "which images are included in chapter 5 of the book OO Programming Language?".

    - **PhysicalObject:** specifies the physical structural unit such as a page or a block, described by attribute *physicalPart.* Hence, it is possible to determine different segmentation types applied to a PersonificationObject. Attribute *metadataContainer* makes it possible to query all properties of page 51 of a report, or to have a copy of page

10 of a paper in English. Similarly, attribute *dataContainer* supports searches such as: "which images are included in page 15 of a book?", or "get the chapter referenced in a certain page of a book".

- **HypermediaObject:** specifies the hypermedia structural unit such as an image, a text, etc., described by attribute *hypermediaType.* Attribute *metadataContainer* makes it possible to get, for example, all the properties of image 3 of a certain paper or the copies of this image in other formats. Attribute *dataContainer* supports queries such as: "in which chapter is image 3 described"?

- **CollectionObject**: as a specialization class of DigitalObject, it is responsible for organizing documents and sub-collections, emphasizing how they are interrelated. Attributes *searchType* and *collectionresourceType* specify, respectively, how collection contents are organized (hierarchically, indexed, etc.) and to which resource category the collection corresponds (a site, a catalog, etc.). Other attributes may be additionally specified here according to specific metadata standards descriptors for collections. Attribute *metadataContainer* supports queries on the complementary references of a collection, on its properties, etc., whereas attribute *dataContainer* provides information about all the sub-collections associated to that collection.

- **AssociationObject**: this object class implements the catalog concept of the Warwick Framework [15], grouping a set of packages under a contextual relationship type which is represented here as a first-class object, having also its own associated metadata. It enables the association of contextual relationships to each of the abstractions levels of a document, which are specified by attribute *associationType* (such as references, terms and conditions for access, bibliographical description, etc.). Attribute metadataContainer recursively references a set of digital objects of AssociationObject type, making it possible to describe recursive relationships. DataContainer references one or more digital objects (collections and documents).

- **MetadataStandard**: through this class it is possible to use proprietary or personalized metadata standards, specifying metadata descriptors for each abstraction level of a document. Once created, these descriptors can be used for describing documents and collections in the repository. These characteristics are specified by the attributes: *standard*, which gives a name to the metadata standard in use; *element*, which identifies the descriptor; and *layer*, which associates the descriptor to the abstract level of a document (content expression, personification or structural). Figure 4 shows an example of how to reference DC standard [5] at the personification level.

- **ContentPackage**: this class stores instances of a document or a collection, according to its abstract level. It can be primitive, when the metadata Package descriptor corresponds to a single value, such as title; or aggregate, when the descriptor corresponds to an aggregation of values such as address, which can be composed of street, number, zip code, etc. It is worthwhile remarking that this class is very relevant in the context of this model: in fact it contains the concrete part conceptually represented in the DocumentObject hierarchy, representing the document organization.

- **DirectAgent**: this class represents entities (persons or organizations) which act directly upon DocumentObject, MetadataObject or CollectionObject. Its subclasses are:

  - **CreatorAgent**: this class represents agents responsible for objects in the repository;

  - **DiverseAgent**: this class represents all agents which interact directly with collections. Attribute type determines, for example, if the agent is the collection producer, the collector, the owner or the administrator of a collection. These agents can have different activities (**ActivityClass**) related to the selling, acquisition, contracts, access delegation, etc. over a collection. A producer, for example, can contract a creator to develop his homepage or can sell a collection for a collector.

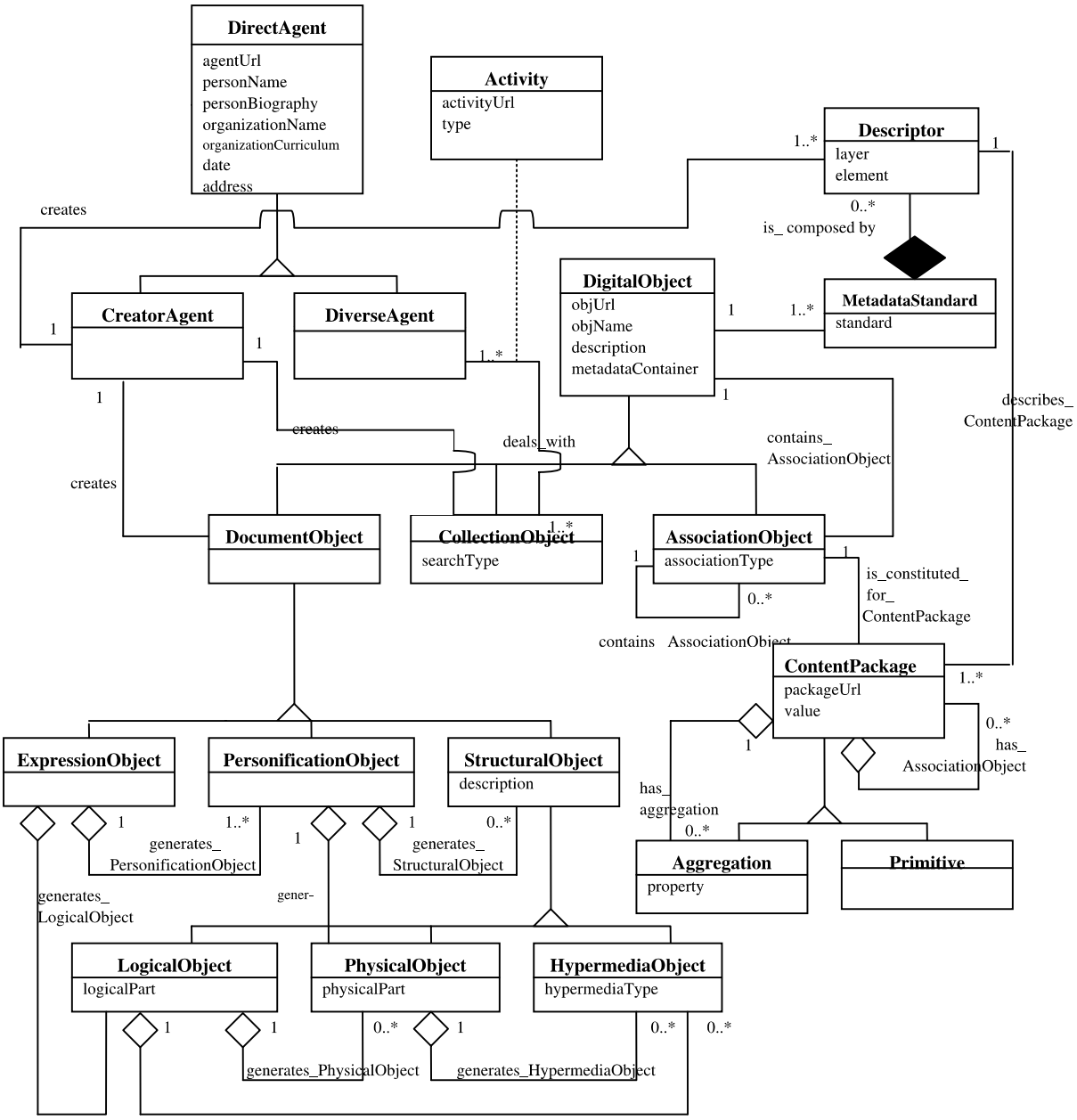


Figure 3: MODDEC schema represented in UML notation

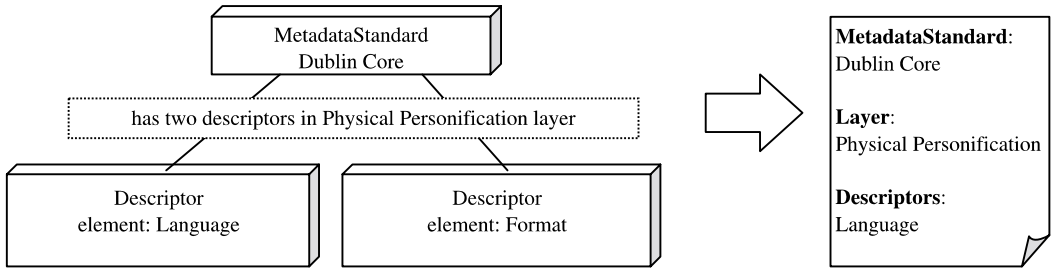


Figure 4: Example of DC standard creation in the personification layer

4 An Example of MODDEC Usage

In order to show the usage of the model, i.e., how digital objects can be described on the Web, we will present an example where different information resources, represented by a node, are linked according to several relationship categories. The schema previously described will be used to represent these associations. Figure 5 presents the description of resource D, a collection object named "Metadata on the Web", with its documents, sub-collections and relationships.

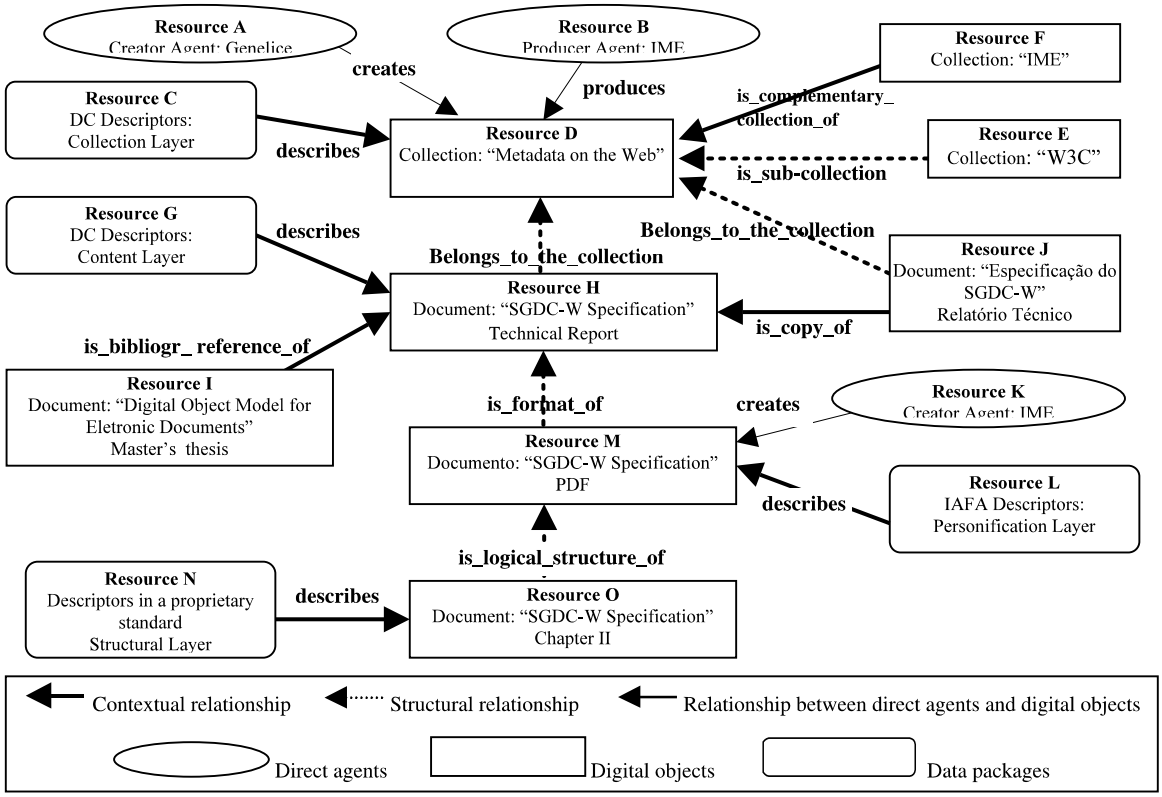


Figure 5: Description of resource D (entitled "Metadata on the Web")

- **Collection Level**: This collection is associated with two direct agents A (creator of D) and B (producer of D) and it has two contextual relationships indicated by its metadata container. These relationships respectively reference the association object Ass1 (describes) and Ass2 (is_complementar_collection_of), as shown in Figure 6. Structural relationships, indicated by data container attribute, reference resources H (identified by Doc1), J (identified by Doc2), and collection E (identified by Col3). Hence, E is a sub-collection of D and J and H are content expressions (technical reports in different idioms) of D. Taking into account that a contextual relationship between two objects requires an object of association type, it is also necessary to specify its source, i.e., a data package. Data container attribute of Ass1 contains the data package (resource C identified by Package1) with DC descriptors describing this collection. Data container attribute of Ass2 has the collection address represented by resource F (identified by Col2), which is a complementary collection of D.

- **Content Expression Level**: Resource H (identified by Doc1), identified in the content expression level, is expressed in terms of a technical report. It has contextual relationships identified by Ass3, Ass4, Ass5, as presented in Figure 7. Object Ass3 specifies the relationship describes between resource H and the data package of G, identified by Package 2. Ass4 expresses the relationship *copy_of* between resources H and J (Doc2), i.e., J is copy of H. Fi-

nally, Ass5 means that resource I (identified by Doc10) is a bibliogr_reference_of H.

- **Physical Personification Level**: Resource M (identified by Doc3) is described in the personification level because it expresses the format PDF used to create the technical report (resource H). Resource K, corresponding to creator agent, identified by Agent2, represents the agent responsible for the resource M creation. It has a contextual relationship Ass6 and a structural relationship referencing resource O (identified by Doc7). Ass6 object establishes the relationship between resource M and L, a data package identified by Package 3. It describes M using IAFA descriptors, as shown in Figure 8.
- **Structural Level**: Figure 9 presents object O which corresponds to the second chapter of document "SGDC-W Specification" in the structural level. It does not have any structural relationship, but it presents a contextual relationship Ass8, which specifies the relationship *terms and conditions_ for_ access* between resources O and N. The latter is a data package (Package 4) which contains descriptors in a proprietary standard.

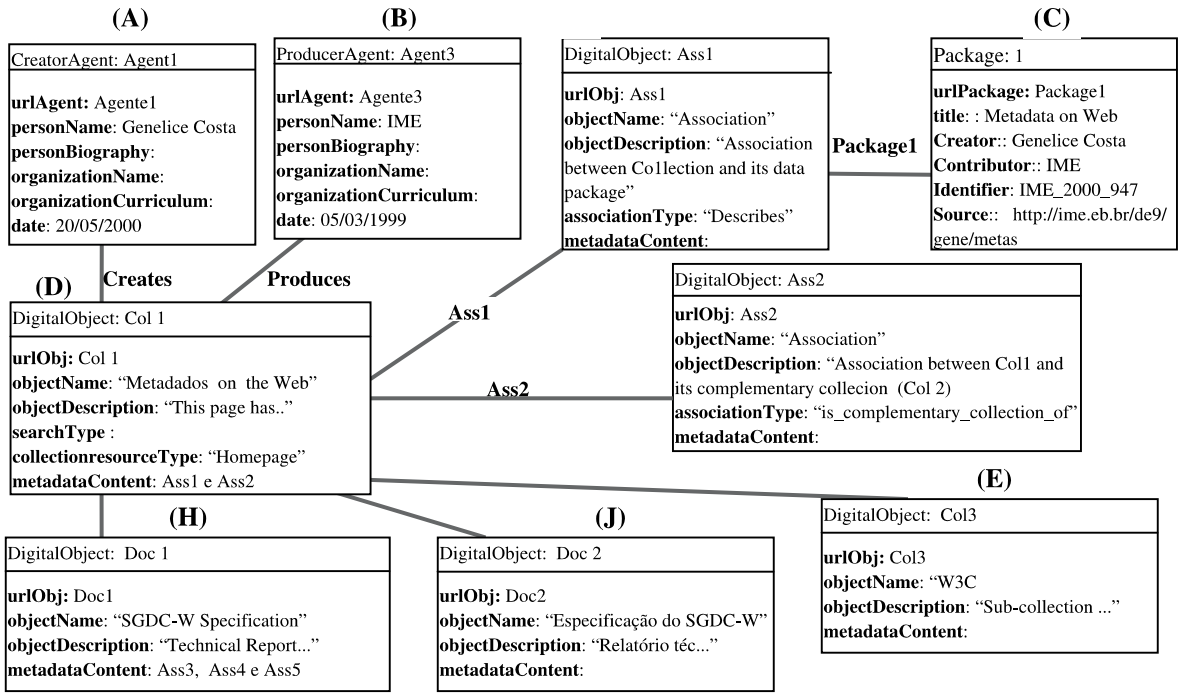


Figure 6: Description of digital object Col 1

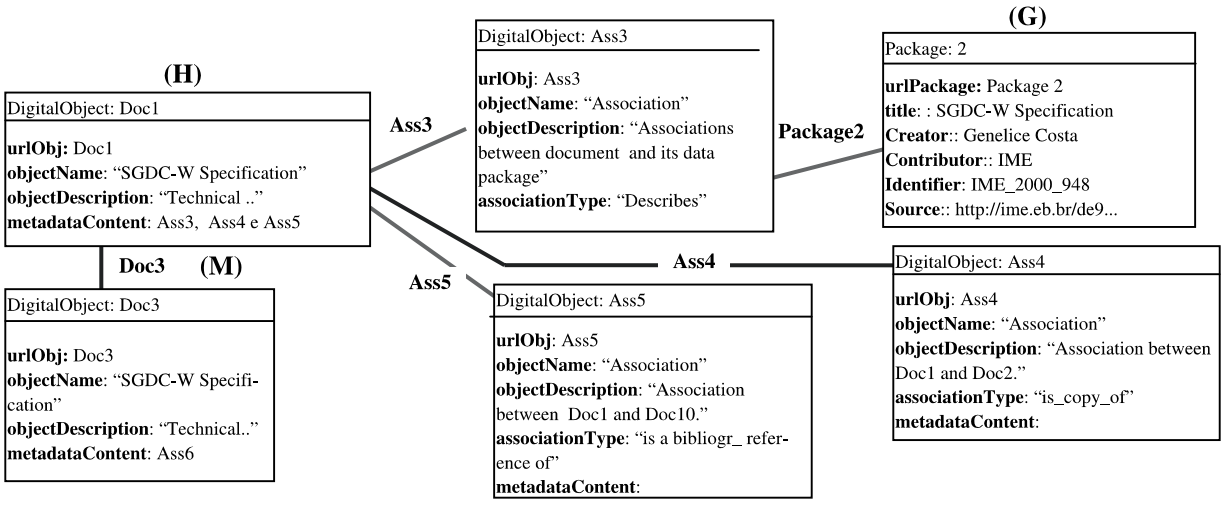


Figure 7: Description of digital object Doc 1

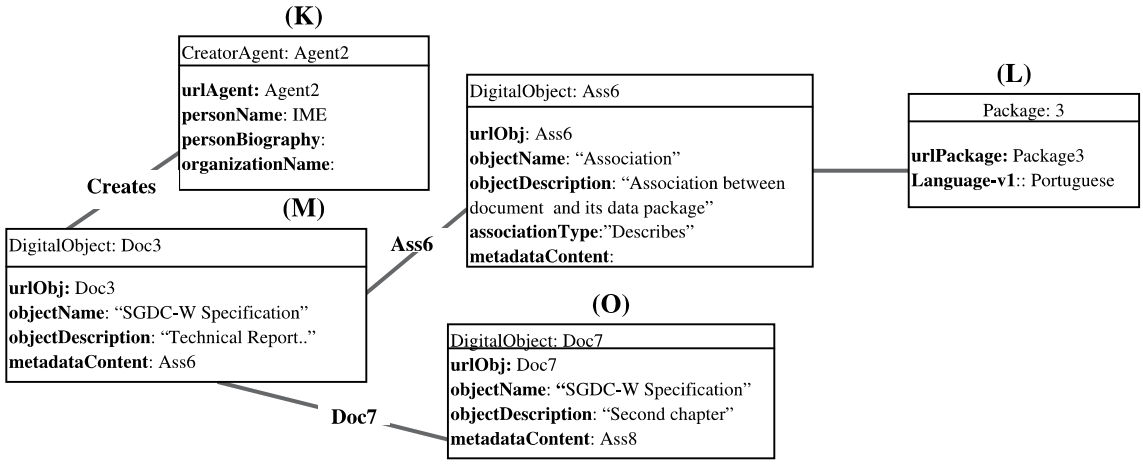


Figure 8: Description of digital object Doc3

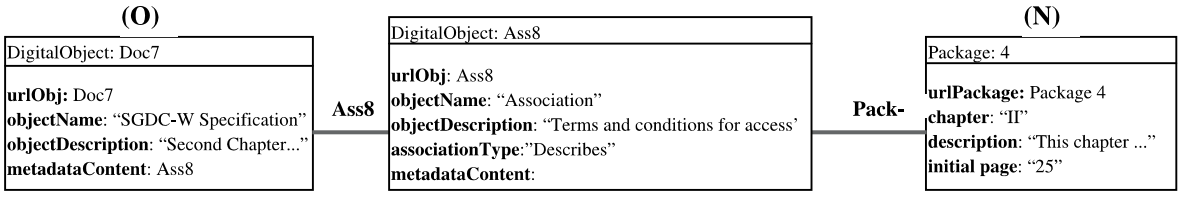


Figure 9: Description of digital object Doc6

5 Developing a Prototype System

As presented in section 3.1, MODDEC is a metadata conceptual model that provides the main infrastructure for developing a tool to manage collections and documents on the Web. This tool should support: the creation of specific metadata standards in order to encompass different knowledge domains; the full description of resources based on descriptors from distinct metadata standards; a friendly interface to help users describe their resources before making them available on the Web, supporting their management, location and retrieval; and an interoperable infrastructure which makes it possible to exchange data and metadata descriptors with other Web services, such as search tools, brokers, etc.

In order to evaluate the potential and functionalities of this tool, it was implemented according to two different approaches: a semi-structured approach, providing syntactic, structural and some semantic interoperability of resources (at data and metadata level) using an RDF-XML-based approach; and a traditional DBMS, using a relational-object model. The main features considered in both implementations will be described next.

5.1 The RDF Approach

According to Berners-Lee [2], in order to have a complete semantic view of the Web, computers should be able to access structured collections and a set of inference rules to represent the knowledge embedded in these resources. Hence, three relevant aspects of interoperability should be considered in this context: the first aspect concerns the semantic expressiveness. It is related to the ability of understanding the meaning of each descriptor within the resource and its relationships; the second aspect concerns the metadata structure, where mechanisms to specify the data organization of a resource, types and possible values within types are defined; finally, the third aspect is related to syntax, which provides rules required for transferring elements, i.e., how data and metadata should be encoded in order to be transferred. W3C [26] has many proposal initiatives to enhance resources exchange on the Web. Combined with XML, RDF [21] turned out to be a powerful mechanism to provide description facilities and semantics of resources in this decentralized environment.

RDF is based on an abstract data model that defines

relationships between resources (pages, documents, person, institution, etc.) on the Web. It represents resources as semantic graphs and uses XML as the syntax language to transport data. It is constituted of a basic RDF which defines elementary entities like resources, properties and statements. Resources are related to others via properties and these relationships are called statements, as shown in Figure 10.

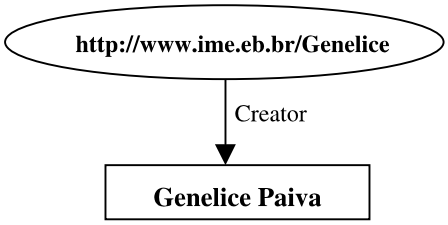
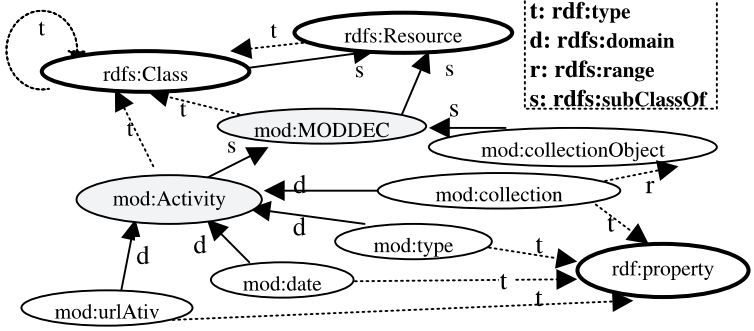


Figure 10: A statement declaration

In this example Genelice Paiva is the creator of resource http://www.ime.eb.br/Genelice. In order to disseminate the use of RDF, W3C has also created RDF descriptions oriented to specific domains, named RDF Schema-RDFS [23]. It offers a distinct vocabulary defined on top of RDF to support the modeling of objects at a conceptual level. It is constituted by three relevant terms: the rdfs:Resource, whose subclasses are rdfs:Class (denoting all classes of the schema) and rdfs:Property (denotes relationships, subproperties and hierarchies between classes). When defining a domain-specific schema in RDF, the classes and properties represented in this schema will become instances of these two resources. Properties allow to specify relationships, hierarchies (rdfs:subClassOf), subproperties and restrictions associated with properties (domain and range restrictions).

According to RDF terminology, Figure 11 presents part of the MODDEC model expressed as a RDF graph and the corresponding code expressed in RDF notation, considered here as a RDF schema. It is worth noticing that values for the properties and classes defined can also be expressed using the same notation. From this representation it was possible to query the generated schema, taking into account the uniform ability to treat data and metadata [12].

a) Moddec expressed as a RDF graph



b)Moddec expressed in RDF notation

```
<?xml version = 1.0"?>
<rdf:RDF
   xmlns: rdf="http://www.w3.org//1999/02/22-rdf-syntax
   -ns#"
   xmlns:rdfs="http://www.w3.org//2000/01/rdf-schema#"
   xmlns:mod="http:// www.ime.eb.br/genelice/moddec#"
/>
<rdfs:Class  rdf: ID = "MODDEC">
<rdfs:comment>Class for representing Digital Objects
   Model and Collections on  Web </rdfs:comment>
   <rdfs:subClassOf rdf:resource = "http:// www.w3.org/
   2000/01/ rdf-schema#Resource"/>
```

Figure 11: Part of MODDEC model expressed as a: a) RDF graph and b) RDF notation

5.1.1 Querying RDF Instances

As already emphasized in last section, instances of RDF applications can be queried according to metadata or data view. In the data view approach RDF descriptions are treated as relational DBMS or XML DTD, and hence the RDF potentiality as data and metadata model is not exploited. XML query declarative languages such as XML-QL, Lorel, XML-GL, XSL, XQL, etc. provide expressiveness for XML queries, however they do not support the RDF schema and hence, the description semantic is lost [3].

Tests using the QueryView tool [27], which applies XQL query language have been successful to provide structural data such as "List all the collections defined in the system", expressed by: MOD/ObjetoDigital//@ident. Symbol '//' provides the ability to navigate through the digital object hierarchy, giving the following result list:

```
http://mirrored.ukoln.ac.uk/dc/_00009
http://info.webcrawler.com/mak/proj/iafa/iafa.txt_00010
http://sunsite.dk/RFC/rfc/rfc1807.html_00011
http://www.domain.com.br/clientes/genelice_00086
http://www.ime.eb.br/de9/espec1.htm_00124
```

In order to explore the semantic provided by the RDF schema generated from an application, a RDF query language should support the main OO concepts (such as hierarchies, classes) and to have access to the schema

vocabulary schema used in RDF [4, 7]. RDF Query [21] is an example of such a language, which explores the metadata view of RDF. It is more complex than SQL, as it makes it possible to query RDF collections containing resources of different types and properties [16]. In what follows we give some examples of queries using this language:

(i) **Selection**: Select all the resources included in the collection http://www.ime.eb.br_00094, having objName as attribute and the string IME.

(ii) **Projection**: Give the e-mail, name and institutions of resources whose Type property has the value Proprietary.

(iii) **Composing** results using Union operator: List the collections and their corresponding documents included in the content expression level.

Figure 12 presents the respective formulations for each of these queries.

(i)
```
<rdfq:rdfquery>
    <rdfq:From
    eachResource="http://www.ime.eb.br_00094"/>
        <rdfq:Select>
            <rdfq:Condition>
                <rdfq:equals>
                    <rdfq:Property
                    name="objName"/>
                    <rdf:String>IME</rdf:String>
                </rdfq:equals>
            </rdfq:Condition>
        </rfq:Select>
    </rdfq:From>
</rdfq:rdfquery>
```

(ii)
```
<rdfq:rdfquery>
    <rdfq:FromeachResource="http://www.ime.eb.br/de9/sgdc-w/
    agents_div/">
      <rdfq:Select propoerties= “email personName organName”>
        <rdfq:Condition>
          <rdfq:equals>
            <rdfq:Property name= "type" />
            <rdf:String>Proprietary</rdf:String>
          </rdfq:equals>
        </rdfq:Condition>
      </rfq:Select>
    </rdfq:From>
</rdfq:rdfquery>
```

(iii)
```
<rdfq:rdfquery>
    <rdfq:Union>
        <rdfq:From eachResource="http://www.ime.eb.br/de9/sgdc-w/obj/colecao” />
            <rdfq:Select>
            <rdfq:Property name="datacontainer"/>
            </rdfq:Select>
        </rdfq:From>
        <rdfq:From eachResource="http://www.ime.eb.br/de9/sgdc-w/obj/documento” />
            <rdfq:Select>
            <rdfq:Property name="ident"/>
            </rdfq:Select>
        </rdfq:From>
    </rdfq:Union>
```

Figure 12: Examples of queries expressed in RDF Query

5.2 The Database Approach

The SGDC-W tool (Management System of Documents and Collections on the Web, in Portuguese) has been developed using this approach. Users interact with the system through a browser and data objects are stored into DB2 UDB [10], an Object-Relational DataBase (ORDBMS). Data can be retrieved using SQL and XQL as well, since a XML Web-server is used, keeping DB descriptions as XML structured documents. The use of XML in this context is justified as an important tool to provide DB information exchange. The system has achieved a good portability due to its implementation using Perl language, allowing it to be installed in different platforms and databases.

Figure 13.a presents the main SGDC-W window whose main functionalities are: Agentes (agents), to provide the inclusion of all kind of agents, such as creator and diverse; Padrões (standards), to enable the creation of metadata standards with their descriptors; Coleções e Documents (Collections and Documents), to allow users to insert their resources using any metadata standard registered in the system. It also makes it possible to establish structural and contextual relationships between digital objects; Atividades (activities), to establish activities between collections and agents; Consultas (queries),

to allow users to have information about the system management or about digital object metadata; and XML document to generate DB instances in XML.

5.2.1 Querying SGDC-W

This module is oriented to querying both the MODDEC structure or to retrieving resources taking into account their metadata descriptors. Consider, for example, a query where a user wants to *retrieve all documents of the collection "Metadados na Web", expressed in the Content Expression level*. Figure 13.b shows this query result containing: the resource URL(which can be accessed directly by a double click); the document name and its description. Other kinds of searches comprising structural and contextual relationships between digital objects can also be formulated such as: *give all the references provided for a certain document; list all papers included in Sigmod Record journal published in April 2001; which are the formats available for the document Enabling Inferecing by Guha?; get all the pages of chapter 4 of the book "How to program in Perl"*.

Figure 14 presents a search in which metadata descriptors from any metadata standard are selected as query conditions. First the user chooses a metadata standard corresponding to each descriptor (if they are not the same), and then the document category (document typology, digital format or document structure). The window containing the selected standard with its descriptors appears, allowing the user to compose his search term condition, using And (E) or OR (OU) operators. In this example the user wants resources containing Metadata as title and Genelice as creator (both from DC). We could also include, for example, the type descriptor (RFC1807) = PDF, as another condition for this query.

5.3 Comparing Implementation Approaches

The RDF approach provides some semantic, syntactical and structural interoperability. However, in the specific case of MODDEC, for which a RDF schema was generated, we observed a semantic loss due to the mapping between UML classes and the RDF schema. Hence, the generated instances can include some inconsistencies. Furthermore, despite important initiatives created by W3C to enhance semantic on the Web, the number of tools (such as parsers and query languages) to support RDF technology is still incipient. Important issues may be raised when using RDF: it is verbose; as it requires Unicode codification, it generates very big XML files requiring more storage capacity; it does not provide data integrity constraints; it has low performance as it does not include DB mechanisms to optimize queries.

On the other hand, the DB approach naturally offers solid query and management facilities. But, on converting from a semi-structured representation to structures of a database management system it falls short on flexibility and semantic expressiveness. Nevertheless, considering the state of the art of today's technology support to the RDF formalism, the solution of combining database and XML technologies seemed to be, for now, the most adequate in the context of this work, even with the semantic deficiencies it contains. Furthermore, this combined solution provides some important advantages, such as [18]: interchange of DB information, which is always consistent according to DB integrity constraints and application business rules; data can be presented dynamically by client configuration using XSL (eXtensible Stylesheet Language), hence providing declarative mechanisms for describing a particular view of the data; data integration from backend DB and other applications; a set of available mechanisms to support XML technology.

a) The main SGDC-W window

b) A query result

Figure 13: The SGDC-W Tool Application

Figure 14: A search with metadata descriptors

6 Conclusion

Adequate metadata management is an essential requirement to provide effective use of information resources on the Web. Based on different frameworks to exchange documents on the Web, regardless of a specific metadata standard, this paper proposed a metadata model to describe and organize documents and collections in this ever-expanding, heterogeneous and distributed environment. Documents are organized into a hierarchical structure, and are described according to metadata descriptors, specific to each level. To the best of our knowledge this is the first work that addresses the problem of organizing resources on the Web using a metadata framework, taking into account associations between resources and their structural composition [20].

This model has been the bases for a tool implemented using both a semi-structured approach, using the RDF model, and a database approach, using the object-relational model. Although RDF model provides a more semantic view of the Web, the existing tools to support this technology are still incipient.

As future work we intend to improve the use of this tool in a more autonomous way, so that part of the metadata can be automatically captured by an agent module, since describing these resources requires considerable time and hard work.

References

[1] C. M. Barreto. A Metadata Model for Describing Electronic Documents on the Web (in Portuguese), Master Thesis, IME-RJ, Aug. 1999.

[2] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web, http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html, 2001.

[3] A. Bonifati, S. Ceri. Comparative Analysis of Five XML Query Languages, Dipartimento di Elettronica e Informazione, Politecnico di Milano, http://citeseer.nj.nec.com/325897-html, 2001.

[4] S. Decker, D. Brickley, J. Saarela, et all. A Query and Inference Service for RDF, http://www.ilrt.bris.ac.uk/discovery/rdf-dev/purls/papers/QL98-queryservice/, 1998.

[5] Dublin Core Metadata Initiative, http://purl.org/DC/index.htm, 2000.

[6] Digital Library Initiative, http://dli.grainger.uiuc.edu/, 2000.

[7] R.V. Guha, O. Lassila, E. Miller et all. Enabling Inferencing, http://www.w3.org/TandS/QL/QL98/ pp/enabling.html, 1998.

[8] R. Guha and T. Bray. Meta Content Framework Using XML, http://www.w3.org/TR/NOTE-MCF-XML-970606, 1997.

[9] David P. Habib and Robert L. Balliot. How to Search the World Wide Web: A Tutorial for Beginners and Non-Experts. http://204.17.98.73/midlib/tutor.htm#GSE, 2000.

[10] IBM. IBM DB2 Universal Database: Application Development Guide: version 6 – Part 4. Object-Relational Programming. 1999. 219 p.

[11] Kansas City Publication Library. Introduction to Search Engines. http://www.kcpl.lib.mo.us/search/srchengines.htm, 2001.

[12] G. Karvounarakis. RDF Query Languages: A state-of-the-art, http://www.ics.forth.gr/proj/isst/RDF/RQL/rdfql.html, 2000.

[13] C. Lagoze, J. R. Davis. Dienst – An Architecture for Distributed Document Libraries. *Comm. of the ACM* 38, 4, Apr. 1995.

[14] LCWEB, Library of Congress, http://lcweb.loc.gov/, June 2000.

[15] C. Lagose, C. A. Lynch and R. J. Daniel. The Warwick Framework – A Container Architecture for aggregating Sets od Metadata. http://www.dlib.org/dlib/july96/lagoze/07lagoze.html, 1996.

[16] A. Malhotra, N. Sundaresan. RDF Query Specification, http://www.w3.org/TandS/QL/QL98/pp/rdfquery.html#jCentral, 1998.

[17] A.M. C. Moura, M.L. M. Campos and C.M.Barreto. A Survey on Metadata for Describing and Retrieving Internet Resources. *World Wide Web Journal*, Vol 1, Baltzer Science Publishers BV, 221-240, Jan. 1999.

[18] MSDN. http:\\msdn.microsoft.com/xml/general/benefits.asp, 2001.

[19] Network Computer Science of Technical Report

Library (NCSTRL), http://www.ncstrl.org, 2000.

[20] G.C.Pereira. Documents and Collections Management System on the Web - SGDC-W (in Portuguese). Master Thesis, IME-RJ, May 2001.

[21] S. Rayavarapu. W3C Query Languages, http://www.coe.neu.edu/~srayavar/W3CQL/ql.htm, 2001.

[22] Resource Description Framework (RDF) Model and Syntax Specification 1.0 - W3C Recommendation 22 February 1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, 1999.

[23] Resource Description Framework (RDF) Model and Syntax Specification – W3C Recommendation 27 March 2000, http://www.w3.org/TR/2000/CR-rdf-schema-20000327, 2000.

[24] L. A. Silva. Dynamic Generation of Digital Library Interfaces Based on Metadata (in Portuguese), Master Thesis, IME-RJ, Jul. 2000.

[25] B. B. Tillet. Cataloging Rules and Conceptual Models, OCLC Distinguished Seminar Series, http://www.oclc.org:5046/~emiller/misc/tillett.html), Jan. 1996.

[26] W3C, http://www.w3c.com/, 2001.

[27] Webmethods. http://xml.webmethods.com/products/QueryView/, 2001.