From Heterogeneous Information Spaces to Virtual Documents

Leonardo Candela, Donatella Castelli, Pasquale Pagano, and Manuele Simi

Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" - CNR,
Via G. Moruzzi, 1 - 56124 PISA - Italy
{candela, castelli, pagano, simi}@isti.cnr.it

Abstract. This paper introduces DoMDL, a powerful and flexible document model capable to represent multi-edition, structured, multimedia documents that can be disseminated in multiple manifestation formats. This model also allows any document to be associated with multiple metadata descriptions in different formats and to include semantic relationships with other documents and parts of them. The paper discusses also how the OpenDLib Digital Library Management System exploits this model to abstract from the specific organization and structure of the documents that are imported from different heterogeneous information sources in order to provide virtual documents that fulfill the needs of the different DL user communities.

1 Introduction

Digital Library Management Systems (DLMSs) [1] are complex systems whose main role is to mediate between content providers and content consumers in order to fulfill information and functionality needs of the DL users.

In particular, DLMSs must support the storage and management of documents collected from heterogeneous information sources. These documents may vary in their structure, format, media, and physical representation. They may be described by different metadata formats and their access may be regulated by different policies. The documents may either be copied from proprietary repositories into the digital library (DL) own repositories or they may be accessed on demand following the link stored into the corresponding metadata records.

The DLMSs must also satisfy the demand of the DL users that want to search, retrieve, access and manipulate documents semantically meaningful in their application domain. Some users for example, may want to see the information space as composed of journals structured in articles, while others may want to work with collections of articles of the same author or with composite documents made by a text and all the images that illustrate that text. Such documents may not correspond to documents submitted to the DL or collected from existing sources, rather they may be virtual documents created by reusing or by processing real documents or parts of them.

Reconciling these two diverse requirements in a complex and application independent framework is one of the most important challenges of a DLMS. In

E.A. Fox et al. (Eds.): ICADL 2005, LNCS 3815, pp. 11–22, 2005.

order to fulfill these requirements a DLMS must be able to abstract from the organization and structuring of the concrete underlying information space and make it accessible through collections of virtual documents tailored to the needs of the DL audience.

This paper focuses on the document model as one of the major factors that influences the level of abstraction provided by a DLMS. In particular, the paper introduces the *Document Model for Digital Library* (DoMDL), a document model which has been successfully exploited in the OpenDLib system [2]. DoMDL can represent multi-edition, structured, multimedia documents that can be disseminated in multiple manifestation formats. This model also allows any document to be associated with multiple metadata descriptions in different formats and to include semantic relationships with other documents and parts of them.

In OpenDLib, DoMDL plays the role of the logical document model that is shared by all the services. Documents harvested from different sources are logically represented to and known by all the OpenDLib services as DoMDL documents. The services provide functionalities that act at the level of abstraction specified by this model. For instance, the storage service is able to accept multiple editions of the same document and automatically generate additional new metadata formats, whereas the index service can index any of the metadata formats associated with an edition. By exploiting the richness and flexibility of the DoMDL model, OpenDLib is thus able to provide the DL users with different views of the information space as this space is populated by virtual documents that fulfill the needs of different application frameworks.

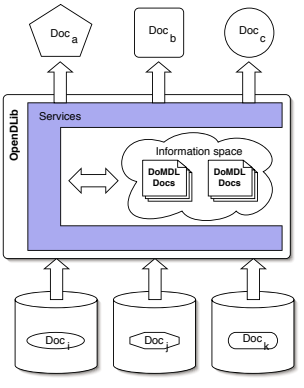

Fig. 1. DoMDL in a DLMS

The rest of this paper is organized as follows: Section 2 presents the DoMDL document Model; Section 3 illustrates how this model has been implemented in the OpenDLib system and how it impacts on the provided functionalities; Section 4 presents two examples, extracted from real OpenDLib DL applications, that show how DomDL has been instantiated to represent specific types of documents; Section 5 compares DoMDL with the document models supported by two other well known DL systems; and finally, Section 6 concludes.

2 The Document Model for Digital Library

DoMDL has been designed to represent structured, multilingual and multimedia documents and can be customized according to the DL content to be handled. For example, it can be used to describe a lecture as the composition of the teacher presentation together with the slides, the video recording and the summary of the talk transcript. However, the same lecture can be disseminated as the MPEG3 format of the video or the SMIL document synchronizing its parts.

In order to be able to represent documents with completely different structures, DoMDL distinguishes four main aspects of document modeling and, using terms and definitions very similar to the IFLA FRBR model [3], represents these aspects through the following entities: Document, Edition, View, and Manifestation (see Figure 2).

The *Document* entity, representing the document as a distinct intellectual creation, captures the more general aspect of it. For example, the book "Digital Libraries and Electronic Publishing" by W. Arms or the lecture "Introduction to Mixed Media Digital Libraries", by C. Lagoze, can all be modeled as Document entities. Each entity of this type is identified via the *Handle* attribute.

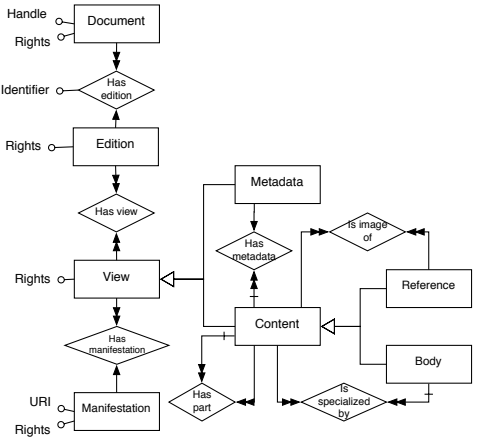

Fig. 2. Document Model for Digital Library

The *Edition*, representing a specific expression of the distinct intellectual creation, models a document instance along the time dimension. The preliminary version of this paper, the version submitted to a conference, the version published in the proceedings, are examples of editions of the same document. These Editions are related to the appropriate Document with an *Identifier* whose value is linear and numbered.

The *View*, modeling a specific intellectual expression, is the way through which an edition is perceived. A view excludes physical aspects that are not related to how a document is to be perceived. For example, the original edition of the proceedings of a DELOS workshop might be disseminated under three different views: *a*) a "structured textual view" containing a "Preface" created by the conference chairs, and the list of thematic sessions containing the accepted papers, *b*) a "presentation view", containing the list of the ppt slides used in the presentations, and *c*) a "metadata view", containing a structured description of the proceedings.

The *Manifestation* models the physical formats by which a document is disseminated. Examples of manifestations are: the MPEG file containing the video recording of the lecture made by C. Lagoze at a certain summer school, the AVI file of the same video, the poscript file of a lecture given by another teacher at the same school, etc. Physical formats are accessible via *URIs*, used to associate local or networked file locations.

These entities are semantically connected by means of a set of relationships. The relationships *Has edition*, *Has view*, and *Has manifestation* link the different aspects of a document. Note that these relationships are multiple, i.e. there can

be several objects in the range associated with the same object in the domain. This means that there can be multiple editions of the same document, multiple views of the same edition and multiple manifestations of the same view.

The View entity is specialized in two sub-entities: *Metadata* and *Content*. The former allows a document edition be perceived through the conceptualization given by its metadata representations. These may be a flat list of pairs (fields, values), as in the Dublin Core metadata records [4], or more complex conceptual structures, such as in the IFLA-FRBR records. Typically, this metadata view is indexed to support attribute-based querying and browsing operations, but it may otherwise be used. For example, it may be disseminated free of charge while the document contents are regulated by fee access, or disseminated on a mobile device. By using the *Has metadata* relationship it is possible to model the fact that also content views can be described by one or more metadata records in different formats.

The Content view has two sub-entities: *Body*, and *Reference*. The former is a view of the document content when it is to be perceived either as a whole or as an aggregation of other views. For example, a textual view of the proceedings a DELOS workshop is built as the aggregation of the textual views of its component articles. The relationship *Has part* links a Body view with its component views. A Body view may be specialized by other views that represent more detailed perceptions of the same content. For example, an article of the cited proceedings may be specialized by two views related to the French and English version of that document, respectively. A view is related to all its specializations through the relationship *Is specialized by*.

The Reference entity represents a view that does not have associated manifestations because it is linked with an already registered manifestation. This entity has been introduced to represent the relationship between views of different document editions. Articles presented at the same workshop, for example, can be modeled as single documents and grouped together by the workshop proceedings document that contains only the references to them. It is worth noting that this entity, bringing together parts of real or virtual documents, makes it possible to manage virtual documents that are not explicitly maintained by the storage system. For example complex reports, or training lectures, can easily be modeled as composition of parts extracted from real documents. A reference view is linked with another view via the relationship *Is image of*.

Each of the entities described above has a set of attributes that specify the rights on the modeled document aspects. This makes it possible, for example, to model possibly different rights on different editions, different access policies on different views or on different parts of the same view, and so on.

3 The OpenDLib Implementation

The document model we have presented has been successfully validated by the OpenDLib experience. OpenDLib [2] is DLMS developed at ISTI-CNR; at present, this system is running in a number of instances serving different institu-

tions. OpenDLib flexibility and customizability are mainly due to the adoption of the DoMDL as logical model to represent both the physical and semantic structure of the documents.

The four ways to think of relationship between a DLMS and its document model are in the following patterns: *(i)* document storage, *(ii)* document discovery, *(iii)* document access, and *(iv)* document visualization. In the following, we first describe the OpenDLib representation of DoMDL and then we address some issues that affect the system from the point of view of the patterns above.

3.1 DoMDL Representation

The representation of a document model usually deals with *(i)* the description of the internal relations among document entities, and *(ii)* the management of the related physical parts of each entity. The OpenDLib solution to these problems is to decouple the definition of the document model instance from its real data. With this approach a document is really composed by several files. The instance of the document model for a given document is described in a separate file, named *Structure file*, which is the only mandatory element that must be provided. The goal of this file is to explain the composition and the relations among the other files that compose the document.

The natural way to express such structured data is through an XML document. Therefore, a major design issue was to define an appropriate XML Schema, able to cover all the DoMDL features. XML Schemas provide a standard means to specify which elements may occur in an XML document and in which order, and to constrain certain aspects of these elements. The result of this effort is the DoMDL XML Schema [5]. An XML document validated against this Schema describes a particular edition of a document; main entities (views and manifestations) belonging to a document are represented with tags while relationships among them are expressed by nesting these tags. As well, a number of attributes on the entity tags allows their type and the related behavior be specified. In this way, a Structure file can put together different physical components to form an unique and coherent structured document. Different editions of the same document are not physically linked together, rather they are logically grouped by the storage model in order to obtain a higher flexibility of the system. The storage model, in fact, is able to manage editions as a single entity since they share the same document identifier.

Finally, according to the document model specification, it is also possible to express a set of rules that regulate the rights on the document views via the properties child tag; in this implementation the rights to download, deliver, transcode or display a view may be, or not be, granted.

In the next section we analyze how the DoMDL model impacts on the OpenDLib system design and how OpenDLib exploits the model to offer new functionalities.

3.2 Related System Issues

The adoption of a particular document model involves the system design and implementation at various levels.

Document Storage. The heart of any DLMS is its storage model, that is how the information is maintained in the system. Here we do not argue about the physical storage manager implemented by OpenDLib since, traditionally, a storage model decouples the document model adopted from the underlying technologies used to store documents. Rather we present both the constraints and opportunities that the utilization of DoMDL has introduced in the system.

Primarily, according to the DoMDL specification, the storage model must be able to manage multiple metadata formats for the same document and multiple physical manifestations for the same view of a document. This allows OpenDLib to be enriched with the capability to: *(i)* automatically move from one metadata format to another one using the provided XSLT stylesheet, and *(ii)* automatically migrate from one physical manifestation (e.g. a pdf file) to another one using a provided transformation procedure or configuring the system to use a 3rd party tool. Among the others, two major advantages that rise from these functionalities are *(i)* to make it easy to create new Digital Libraries starting from existing heterogeneous information sources and *(ii)* to preserve documents from the technological obsolescence. Regarding manifestations, they are identified by URIs. A manifestation can be stored inside or outside the system, depending on the time in which the URI is dereferenced. When a new document is submitted, a number of solutions is offered in order to support a range of different needs. In fact, a manifestations can be: *(i)* directly uploaded with the document, *(ii)* automatically retrieved from an external location and locally stored, *(iii)* maintained as an external manifestation and dynamically retrieved at the access time, or *(iv)* maintained as an external manifestation and displayed through its original location at the access time. These options are made available by properly combining the values of the attributes of the manifestation tags in the document Structure file. The combination of these options at document level makes it possible to build new structured documents that enrich the original ones by aggregating multiple parts of different documents from different heterogeneous information sources. Moreover, these choices promote an optimal utilization of the storage resources. For instance, if an manifestation requires too many storage resources to be stored internally, it can simply be referred to its external original location. This optimization is also supported by the reference view mechanism. Following the model specification, a view can be a reference to another view of a different document; by implementing this mechanism, data duplication is avoided.

The last advantage we mention here is the possibility to submit and manage documents that are modeled in very different fashions in the same OpenDLib instance, if they are compliant with the DoMDL XML Schema. This introduces a high level of flexibility and promotes a full integration among heterogeneous information sources with different types of documents or metadata.

Finally, let us mention addressability, i.e. the granularity of documents that can be directly addressed or referenced. The basic addressable unit is the single manifestation. Moreover, the list of all views or manifestations as well as the list of editions of a document can also be addressed.

Document Access. The access granularity, i.e. how a document or its components can be accessed, is closely tied to the storage model. Possible options include: *(i)* to expose data according to the document model representation, and *(ii)* to hide the representation and provide an interface to query the model in order to obtain the document parts. OpenDLib implements both solutions by providing direct access to the Structure file and also an interface to query a given document. This design choice allows users to select the option that fulfils their needs at best. For instance, to speed up the operations, other OpenDLib modules retrieve from the storage subsystem the Structure files and then manage the corresponding documents. External applications should instead request the document entities (e.g. all the editions of a document, all the manifestations, etc.) in order to be independent from the DoMDL representation.

Document Discovering. Documents discovering is a crucial component of any distributed Digital Library system. This feature is usually achieved through indexing and search mechanisms. OpenDLib provides these functionalities both on the document metadata and, when possible, on the documents themselves (full-text indexing) via its search subsystem. The adoption of DoMDL had a great impact during the design of this subsystem because documents can be expressed in any format and thus no assumptions could be made about the presence of any field or structure of the indexed information. The result is a highly customizable search subsystem based on: *(i)* a complete configuration of any index concerning the metadata or manifestation format, the elements to be indexed and the set of elements to be return after a query, *(ii)* an abstraction layer between the query engine and the format-independent query language supported, and *(iii)* inspection mechanisms that support the discovery of which indexed format, which query operators and which result sets are supported by a particular instance of an index. Therefore, thanks to the document model, an OpenDLib instance can have multiple indexes able to index any format independently of their number or location. Also the graphical user interface provided to interact with the search subsystem has the capability to configure itself, depending on which index it currently interacts with, by automatically adding, removing or changing both its components and look and feel. In addition, the search subsystem offers the very new possibility to execute queries across documents handled by different information sources and expressed in different formats.

Document Visualization. The visualization of documents is the last main issue strictly related with the document model. DoMDL gives a great number of opportunities for the presentation of complex documents. For instance, it allows document visualization be personalized by deciding who has the rights to view what.

OpenDLib provides two kinds of document visualization, one tab-based and one window-based, both able to display documents compliant with the DoMDL model. In either mode, a graphical rendering of the document structure is visualized and manifestations are retrieved on demand next to the user requests. As well, OpenDLib can easily be extended with additional visualization features;

this is specially useful for OpenDLib instances that manage classes of documents with the same structure, e.g. papers or talks, to better exploit the specific structure of those documents. The mechanisms above make it possible to present the same document in different ways by making the concept of *virtual document* concrete.

4 DoMDL Exploitation

In this section we present two successful stories of exploitation of the DoMDL features in the context of digital library instances that are powered by OpenDLib. We do not want to show these examples to validate the design choices made by these digital libraries to represent their documents; rather we want to demonstrate that DoMDL is suitable to accommodate the needs of different user communities.

The first example we report is extracted from the DELOS DL [6]. This DL handles documents published by the homonymous Network of Excellence on Digital Libraries. It stores, maintains, and disseminates, among the others, the proceedings of several DELOS events like the ECDL conferences, a number of thematic and brainstorming workshops, and the international summer schools.

These documents are characterized by a large number of inter-relationships that are emphasized to improve the accessibility and readability of semantically related documents.

Figure 3 depicts a typical edition of an ARTICLE maintained in this DL. Each edition has the following views: *Metadata*, *Abstract*, and *Content* which are expressions of the article related to manifestations in different formats; *Related Talk*, which links with the presentation of the article made by its author during the related event; and *In Proceedings*, which links with the document that represents the proceedings where the article has been published. Reference views are also used to link a TALK document with the content of the edition of the respective article. It is also important to point out that in the DELOS DL different metadata formats are used to represent the description of an article, that multiple manifestations in different content types are associated with the same view, and that the video manifestations are stored on video streaming servers able to improve their fruition. Finally, we highlight that the end-user perceives an intellectual creation via the homogeneous and coherent presentation of a virtual document

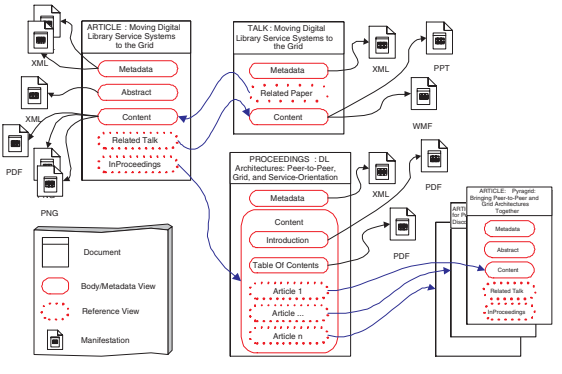

Fig. 3. DELOS Digital Library documents

that, instead, is obtained collecting parts of different and heterogenous stored documents.

The second example is extracted from the ARTE DL [7]. This DL stores, maintains, and disseminates the digitized versions of ancient texts and images linked by relationships that express semantic associations among them, such as the *contains*, *is contained in*, *is related to*, and *has authored by* relationships. The original documents are collected from very heterogeneous information sources: *(i)* ranging from different types of database to file-system based storage systems, and *(ii)* based on proprietary content representations. A typical edition of an ICONOGRAPHY is represented by its metadata and related picture. The value added by using DoMDL is perceived by analyzing the document relationships. Using reference views, it has been possible to model virtual documents allowing end-users to navigate the relationship from an iconographic document to the book that contains it, analyze the textual part before and after the mentioned picture, browse the book to see other similar documents, and also immediately access to the other related iconographic documents.

Finally, it is important to note other specific characteristics of the documents managed by this digital library, namely: the wide heterogeneity of the representation and description formats; the existence of access policies regarding many parts of the documents; and the variety of new documents that are created by the members of the digital library by composing parts of existing documents.

5 Related Works

Most of DLMSs are designed to manage *simple* documents, i.e. documents composed by a single entity having a fixed metadata format. In this section we present a comparison between DoMDL and two rich document models that have been designed to fulfill requirements arising from different contexts, the DSpace data model and the Fedora digital object model.

5.1 The DSpace Data Model

DSpace[1] [8, 9] is an open source digital library system designed to operate as a centralized system for capturing, storing, indexing, preserving, and redistributing documents in digital formats. It has been designed to fulfill the requirements of a university research faculty for managing its intellectual outputs.

Item is the basic archival element in DSpace and thus it corresponds to the DoMDL Document entity. An item is organized into bundles of bitstreams, where a *bundle* is a set of somehow closely related bitstreams corresponding in part to our View entity, while a *bitstream* is a stream of bits, usually is a computer file and it is thus close to the physical part of our Manifestation entity. For example, a document having two different manifestations, a PDF and an HTML one, is

[1] `http://www.dspace.org`

modeled in DSpace with an item having two bundles: *(i)* the PDF manifestation that has a bitstream representing the PDF file, and *(ii)* the HTML manifestation with a set of bitstreams representing HTML files and images that compose the main HTML manifestation.

The ordered sequence is the only type of relationship that can be expressed between bitstreams of the same bundle. Moreover, the concept of edition is not explicit within this model, even if it may be modeled via particular structural metadata by adapting some DSpace components. Further, references, that make it possible to build documents by aggregating already existing ones, are not modeled explicitly.

DSpace manages descriptive, administrative and structural metadata. Regarding descriptive metadata, each item has one qualified Dublin Core metadata record. This schema can be changed but the system search and submission functionalities are not capable to automatically react to these changes and thus they must be updated. Moreover, it is possible to manage just one metadata record for each item, i.e. if there are two or more descriptive metadata records about an item, only one is considered as the metadata record while the others can only be stored into the system by using the bundle concept and thus they will not be used in the discovery phase. Administrative metadata include (a) preservation metadata and (b) authorization policy metadata in a sort of proprietary format. DSpace structural metadata can be considered as being fairly basic, i.e. bitstreams of an item can be arranged into separate bundles as described above, and this probably will be an area of future development as DSpace designers state.

5.2 The Fedora Object Model

Fedora[2][10, 11] is a repository service for storing and managing complex objects. At its core there is a powerful document model and thanks to the richness and flexibility of this model the system is nowadays used by many institutions.

A Fedora digital object is composed by *i*) a *unique identifier*, *ii*) a set of *descriptive properties*, *iii*) a set of *datastreams*, and *iv*) a set of *disseminators*. Descriptive properties are the information needed for the management of the objects within the repository, i.e. the object type, its state, its creation, and last update date. The type is used to distinguish among the primitive Fedora objects while the state is used to distinguish among active, inactive and deleted objects.

Datastreams are containers used to maintain both data and metadata belonging to an object. Thus, the same concept is used to model bytestreams representing the document, as well as metadata to express relationships with other objects, policies and audit data. Moreover, a datastream is either used to encapsulate any type of bytestream internally as well reference to it externally. In this way, on the one hand it is possible to aggregate local content with exter-

[2] `http://www.fedora.info`

nal content, on the other hand there is not a complete decoupling between the document structure and its content. This broad concept of datastream, equivalent to the DoMDL view, uses reserved datastreams to differentiate between its types. For instance, a datastream of type DC is used to express the DC Metadata record [12], while a datastream of type REL-EXT is used to express object to object relationships following a well established ontology of relationships[3].

Disseminators are components capable to associate an external service with the object in order to supply a virtual view of the object itself, or of its datastream content. The Fedora repository, interoperating with the service, is in charge to produce this view. As a consequence this approach is object centric, i.e. in order to create a new view over a set of documents it is needed both to create a service capable to offer it and to update all the objects that this service must act on.

Fedora object model covers also versioning related to components, i.e. datastreams and disseminators. The system automatically creates a new version of them whenever they are modified, while maintaining also the former representation without changing the document structure. Thus, within the same component identified by its identifier, all the versions are maintained and identifiable via their own identifier.

DoMDL and Fedora digital object model have many commonalities and both aim at managing complex structured documents. Main differences are related with the mechanisms for offering virtual views as well as with how they decouple structural information from content information.

6 Conclusion

This paper has illustrated the DoMDL model and discussed how it has been exploited by the OpenDLib system to achieve the ability of abstracting from the specific organization and structure of the documents whether they have been submitted or harvested from exiting heterogeneous information sources. The paper has also shown how the proposed model has been used to support the document representation requirements of two different OpenDLib empowered DLs. The reported examples have highlighted the DoMDL capability of representing complex documents built by aggregating related parts, where each part, in turn, may be shared with other documents. These characteristics, together with the ability to distinguish different aspects of a document and associate multiple matadata formats, are the major distinguishing features of DoMDL with respect to other DLs document models.

We strongly believe that these features will also be important in the future DLs which will support the construction of documents which have no analogous in the traditional physical world. In this framework models like DoMDL will provide the necessary substrate for the semantic layer of a DL information model [13].

[3] http://www.fedora.info/definitions/1/0/fedora-relsext-ontology.rdfs

References

1. DELOS-NSF Working Group on Digital Library Information-Technology Infrastructures: Final report. Technical report, DELOS NoE and NSF (2003)
2. Castelli, D., Pagano, P.: OpenDLib: A Digital Library Service System. In Agosti, M., Thanos, C., eds.: Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 2002, Proceedings. Lecture Notes in Computer Science, Springer-Verlag (2002) 292–308
3. IFLA Study Group on the Functional Requirements for Bibliographic Records.: Functional Requirements for Bibliographic Records: Final Report. (`http://www.ifla.org/VII/s13/frbr/frbr.htm`)
4. Dublin Core Metadata Initiative: Dublin Core Metadata element set, version 1.1: Reference description. (`http://dublincore.org/documents/dces/`)
5. OpenDLib: DoMDL XML Schema. (http://www.opendlib.com/resources/-schemas/domdl.xsd)
6. OpenDLib: The DELOS Digital Library. (http://delos-dl.isti.cnr.it)
7. OpenDLib: The ARTE Digital Library. (http://arte-sns.isti.cnr.it)
8. Tansley, R., Bass, M., Stuve, D., Branschofsky, M., Chudnov, D., McClellan, G., Smith, M.: The DSpace Institutional Digital Repository System: current functionality. In: Proceedings of the third ACM/IEEE-CS joint conference on Digital libraries, IEEE Computer Society (2003) 87–97
9. Tansley, R., Bass, M., Smith, M.: DSpace as an Open Archival Information System: Current Status and Future Directions. In Koch, T., Sølvberg, I., eds.: Research and Advanced Technology for Digital Libraries, 7th European Conference, ECDL 2003, Trondheim, Norway, August 17-22, 2003, Proceedings. Lecture Notes in Computer Science, Springer-Verlag (2003) 446–460
10. Payette, S., Thornton, S.: The Mellon Fedora Project: Digital Library Architecture Meets XML and Web Services. In Agosti, M., Thanos, C., eds.: Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 2002, Proceedings. Lecture Notes in Computer Science, Springer-Verlag (2002) 406 – 421
11. Lagoze, C., Payette, S., Shin, E., Wilper, C.: Fedora: An Architecture for Complex Objects and their Relationships. Journal of Digital Libraries, Special Issue on Complex Objects (2005)
12. DC Team: Dublin Core Metadata Initiative. (`http://dublincore.org`)
13. Del Bimbo, A., Gradmann, S., Ioannidis, Y.: Towards a Long Term Agenda for Digital Library Research. Third Delos Brainstorming Meeting Report - Corvara - Italy, DELOS NoE (2004)