Designing an Integrated Digital Library Framework to support Multiple Heterogeneous Collections

G. Pirounakis, K. Saidis, M. Nikolaidou, E. Lourdi
Library Computer Centre, University of Athens
Panepistimiopolis, 15771Athens, Greece
Email: mara@di.uoa.gr

ABSTRACT
Athens University recently initiated a digital collection development project to provide enhanced educational capabilities. Collections vary in terms of the material included and the requirements imposed by potential users. Collections are developed by cataloguers and researchers working in specific university libraries, while all of them are administered by the Libraries Computer Centre. In order to simplify collection management and promote collection interoperability, a common digital library platform should be employed to support all collections. To deal with the extended requirements imposed, it was decided to extend an open source digital library system, rather than using "out of the box" software. In this paper, we present the criteria used to choose a DL system for developing the University's digital collections and discuss the prototype version of the system built using Fedora and DSpace. In order to evaluate them, the Folklore Collection was used as an example. Conclusions drawn from their comparison and the proposed integrated DL architecture based on Fedora are also presented.

KEYWORDS

Digital library architecture, Describing and managing collections, Fedora, DSpace.

1. Introduction

Athens University recently initiated a digital collection development project to gather research material produced by its members and provide enhanced educational capabilities. Each collection has specific scientific or cultural significance, includes different types of material (e.g. text, music, photographs, videos, scanned documents), consists of either digital or digitized material and satisfies diverse user requirements in terms of object structure, metadata and presentation. Most cultural collections are archival in nature and contain digitized material, while most scientific collections are constantly updated and contain digital material. Collections are developed by cataloguers and researchers working in specific university libraries, while all of them are administered by the Libraries Computer Centre (LCC). In order to simplify collection management and promote collection interoperability, it was decided to employ a common digital library (DL) platform to support all collections. Requirements imposed for the digital library platform involve:

1. The material added in the digital library system must be characterized by educational and specialized research properties useful for specific research areas. Metadata management becomes more complicated, as combinations of different metadata schemes, such as Dublin Core [Dublin Core Metadata Initiative], IEEE Learning Object Metadata [IEEE, 2002] or even local fields may be used to describe collections. Collection level metadata should also be supported.
2. The digital library should support more than one collection. Thus, specific features must be provided for individual collections, determining the type and structure of the digital material. Although a core metadata element set can be identified, the metadata used to describe collections differ. To facilitate multiple collection search, the definition of crosswalks between metadata schemes should be supported [Yu, 2003].
3. The number and nature of supported collections are not predefined. Thus, designing an overall DL architecture, we need to consider that dynamically creating and administering collections is required.
4. Interaction with a new collection in the Digital Library should be seamlessly supported. Thus, existing services should be functional for all collections without additional programming effort.
5. The digital library environment should support a complex and parameterized workflow, as research material may be added directly by the researcher, while he/she also participates in metadata creation. Furthermore, the same workflow application should be used for all collections.
6. All features provided need to be bilingual. Both Greek and English languages are supported in material characterization and presentation.

To deal with the extended requirements imposed, one should select the appropriate DL software and properly extend its features. The basic prerequisites of this software are to be open-source and provide interoperable features (OAI-PMH, open standard file types, public APIs) It should also provide preservation solutions and manage digitized content. Libraries Computer Centre policy emphasizes in extending a fully customizable digital library system, than using "out of the box" software, thus one of the basic requirements is to support an open modular architecture.

A list of open-source institutional repository software is proposed in [Open Society Institute, 2004]. In order to evaluate them, the *Folklore Collection* was used, since it is a typical example of a complex collection and conclusions drawn from its support may be applied in other collections as well. The main characteristics of Folklore collection are discussed in section 2. The criteria used to choose a DL system from those proposed by Open Society Institute [Open Society Institute, 2004] are presented in section 3. Two of them were selected for evaluation: *Fedora* [Staples, 2003] and *DSpace* [Smith, 2003]. These two systems are consistent to all basic requirements and already have a large number of installations worldwide.

In order to evaluate both systems, a prototype version supporting the *Folklore Collection* was built using them. The remarks drawn from their comparison lay in section 4. The integrated DL architecture designed to support Folklore collection is described in section 5. It is based on Fedora system. Conclusions reside in section 6.

2. Folklore Collection Characteristics

The folklore collection supported by Athens University is dedicated to local tradition and customs of several regions in Greece representing the way of living and thinking in these regions through the last two centuries. The folklore collection consists of travelling notebooks written by the students of the Greek Literature Department. They are composed by notes and maps created by the author and lyrics or handcrafts related with a specific region. The lyrics and handcrafts included in a notebook must be treated both as parts of it and as independent objects belonging in a different sub-collection. Specifically the folklore collection is divided into:

a) Notebook sub-collection. Each notebook is written by a student after local research and refers to a specific area or village. The notebook is separated into predefined chapters and subsections and includes a table of contents. Most of the notebooks are accompanied by drawn maps, photographs of habitants and regions, artifacts (e.g laces or doles) and sound recordings with songs and folk music.
b) Photographs sub-collection, consisted of the photographs that inside the notebooks as accompanying material
c) Artifact sub-collection, exposed in the library and
d) Sound recordings sub-collection, consisting of local music, lyrics and tale recordings related to the notebooks.

In order to support the folklore collection, the following requirements should be satisfied:

Sub-collection support
Due to the variety of material and the complex relations between folklore collection resources, the collection must be organized into sub-collections by unifying kindred resources into groups according to material type. By organizing the folklore collection into sub-collections, the attributes inherited from the collection to sub-collections are identified and the overall collection can be easily navigated by users using specific access points like the date or the topic of coverage.

Collection-level description and definition
High-level collection description is important in order to help the navigation, discovery and selection of cultural content. For the same reason, it is required to offer a detailed collection and sub-collection level description with the appropriate metadata elements, after specifying the structure of the collection. The collection description provides information about the contents and size of the folklore collection, about the purpose and the historical context in which the collection or sub-collections have been created and elements for the administration, the technical requirements and the structure of the collection. Specifically, the folklore collection is described by the Dublin Core Collection Description Application Profile, while the schema is extended by local elements related to custom properties (for example marriage, local food specialties, eating customs).

Representation of compound objects
Every notebook contains written text separated into predefined chapters and subsections and a table of contents. The headings of chapters and subsections are predefined and correspond to specific aspects of every day life, as dressing code, eating customs, entertainment and religious customs. A list of headings is provided to all students although they are not obliged to select information for all of them. Thus, the structure of notebooks is predefined. Each notebook is accompanied by artifacts, photographs and maps. In its written form, it is difficult to search for information inside the notebook and in order to do this, researchers must most times read the whole notebook. Notebooks must be represented as compound digital objects having separate parts (chapter and sub-sections), which should be characterized individually. Thus, metadata should be kept in notebook, chapter and sub-section level, while specific metadata fields should be inherited to the lower-level entity.

Description of existing relations
It is necessary to represent all kinds of relations that exist inside and outside the collection through all the structural levels, in order to provide the users with all the information that is hidden in the collection. For example, the relation between the photograph referenced at a notebook page and the actual photograph belonging in the photographs sub-collection probably in another format should be identified (for example: “has format” or “is converted to” or “is the same with”). In this way, while the user reads the notebook, may also view the photograph and similar ones by browsing the photographs sub-collection.

Appropriate metadata support
Due to the variety of folklore collection material, different metadata schemes, as DC and LOM, and local fields should be supported. This is a strong necessity in order to keep all the valuable information for preservation, authenticity and retrieval of information. It is important for the users to have many access points to the content of the folklore collection and to be able to search by date, subject, geographic domains and the type of objects. DC is adopted as the basic metadata scheme, while it is further extend to cover specific aspects as i) access rights in order to protect the copyright of the oral tradition, ii) elements for the varied audience that the folklore material covers, iii) the educational character and the purpose of every resource and iv) the technique of digitization and the technical requirements.

3. Criteria for Selecting a Digital Library System

This section discusses the DL System requirements imposed by LCC. These are divided into two main categories; each of them identifying features related either to the storage of information, either content or metadata, or the design and implementation of the DL system. More specifically, the selection criteria adopted are the following:

Storage Capabilities / Collection Issues

- Basic Preservation capabilities: The DL System should handle effectively preservation issues, by assigning persistent unique identifiers to digital objects and providing support for various file formats and versions for the storage of their content.
- Multilingual support: The system should support at least the Greek and English languages, with regard to both content and metadata storage and presentation.
- Effective Support for Digitized Content: As illustrated by the description of folklore collection, the system should provide the ability to handle effectively digitized content. Moreover, the majority of the Libraries collections considered for digitization in the future are sharing the similar dependence on digitized content support with the folklore collection (Byzantine music manuscripts and records, etc).
- Support for multiple, heterogeneous collections: The point described above depicts the requirement for the efficient handling of multiple, heterogeneous collections.

Design / Implementation Issues

- Interoperability support (OAI-PMH, XML, public APIs): Interoperability between the University's digital library and similar systems in Greece or worldwide is an issue that will eventually occur, as our experience has shown. By identifying it as a distinct requirement of the DL system we aim to the inclusion of interoperability issues in the design and development of the integrated DL platform right from its beginning.
- Flexibility and Expandability: The system should be flexible and extensible, allowing the addition of extra functionality in a straightforward manner. This issue suggests that the selected DL System should impose minimum restrictions regarding its usage patterns and scenarios.
- Separation of content storage and representation / interfaces: The system should separate the representation logic from its core repository / storage functionality in the highest possible degree. DL service must be included into an integrated web environment, the LCC portal, so it is of great importance to be supplied with the ability to "program" the interface in arbitrary ways while the DL system handles the storage, preservation and content retrieval issues independently.
- Implemented in Java: The development of LCC portal has already been started using Java to integrate backend databases, the OPAC and collaboration software used by Library staff for every day activities. By requiring the implementation of the DL System to be in Java, digital collection will be integrated into the LCC computing environment in a natural way.

The above requirements highlight the main DL System selection criteria. No existing digital library system could be used "out of the box" to implement folklore collection, or other collections of the same diverse and complex nature. In order to develop an effective, usable and integrated digital library platform, the University is focusing on settling a long run investment on the selected DL System. Under this perspective,

another important requirement is added, which refers to the ability to freely extend and customize the selected system. In other terms, the selected DL System should be distributed under an Open Source License.

4. Digital Library System Comparison

A list of open-source institutional repository software is proposed in [Open Society Institute, 2004]. Based on the criteria presented in the previous section, two of them were chosen for further evaluation: *Fedora* [Staples, 2003] and *DSpace* [Smith, 2003]. These two systems are consistent to all basic requirements and already have a large number of installations worldwide. Fedora and DSpace are based on open and modular architectures. The first one is using Flexible and Extensible Digital Object and Repository Architecture (FEDORA), while the second is based on a three-layered architecture and a data model influenced by the Open Archival Information Systems (OAIS) reference model [CCSDS, 2002]. The main modules of each system provide public APIs to access and manage metadata and digital content. Both of these systems support preservation issues, by providing many digital formats of the same content, using technical metadata and retaining a global unique identifier to access each digital object. They support digitized objects, more than other platforms that are oriented on born-digital content, mainly electronic documents. The systems are not restricted to specific file formats or digital content type (documents, photographs, sound or videos). In order to evaluate both systems, a prototype version supporting the *Folklore Collection* was built using them.

4.1. Fedora

Fedora is a java based open-source digital repository system comprised of a flexible and extensible architecture. The basic entity of Fedora repository is digital object. A digital object is comprised of a persistent identifier (PID), system metadata, one or more datastreams and disseminators that associate datastreams to behaviors. Datastreams are used to represent metadata or digital content. Digital objects are stored internally as XML files based on an extension of Metadata Encoding and Transmission Standard (METS) schema. There are three distinct types of digital objects: data objects, behavior definition objects and behavior mechanism objects. The first one represent entities that contain the content and metadata, while other two define and implement the methods that present or transform the content of digital objects. It provides two public APIs for the management services (API-M, API-M-Lite) and two for access services (API-A, API-A-Lite). APIs are implemented as SOAP and HTTP-enabled web services [Payette, 1998]. First version of Fedora became available on May 2003 under Mozilla public license and the current version is 1.2. Fedora is flexible and easily customized to support folklore collection specific features.

Collections are not natively supported by Fedora. In order to describe collections, it is practical to use a data object to represent each collection containing the appropriate collection description and rights metadata and the templates for the creation of data

objects. Sub-collections can be defined in the same way and the relation between the parent and child collection can be described by specific structural metadata in both data objects. Sub-collection objects can inherit description and rights metadata from the parent collection. Additionally, a collection management module should be implemented that will communicate with the management API and control the aforementioned functionality.

A composite physical object, such as folklore notebook, can be represented as a number of data objects. Some of the data objects represent logical entities of the physical object, as chapters or the whole notebook view. Others represent the physical entities, for example the pages of the notebook. The relations between those data objects comprise the structure of the composite physical object.

Relations are necessary to represent the structure of composite objects or to relate independent data objects that belong to the same or different collections. To support relations in Fedora, a special datastream can be used on each data object that will contain the structural metadata of it. A behavior object can be associated to the data object and describe the methods that will represent the relations in presentation level. These methods must be implemented in a general manner in order to support each relations special requirement. An extension module must also been developed over Fedora management API in order to manage the relations between data objects.

Every metadata model can be described and accessed in one or more datastreams of the digital object. The metadata model can be a local metadata set, a standard metadata set or an extension of Dublin Core metadata element set. The disadvantage is that Fedora supports indexing and searching services, only for Dublin Core metadata element set, so an external application should be used to index other metadata fields.

4.2. DSpace

DSpace is an open source digital repository system, implemented in Java and primarily focused on institutional and research material (reports, research papers and publications). DSpace provides an “out-of-the-box” solution for the problem of collecting, storing, preserving, indexing and distributing such material in digital form. It is developed by MIT and Hewlett Packard, released under a BSD open source license and its most current version at the time of this writing is 1.1.1.

DSpace elaborates on a typical and straightforward three-tier architecture, consisting of a storage layer, responsible for the storage of items (digital objects) and their metadata using the PostgreSQL relational database. Digital content (files) is stored in the file system, and associated to items in terms of bitstreams and bundles allowing an item to contain various files. Business logic layer consists of a numerous components handling individual aspects of the DSpace system, such as browsing, searching (based on Apache Lucene), user/group management and authorization, workflow management, content management and administration. Finally, the application layer provides the end user interaction and interface functionality, in terms of web user interface, batch item importing facilities, OAI metadata providers and the like. DSpace uses a qualified Dublin Core metadata scheme for describing items and provides the ability to expose them for interoperability reasons, through the Open

Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). A significant restriction refers to the fact that its export facilities do not abide to METS, but this extension is under development. Regarding preservation issues, DSpace provides support for CNRI handles, assuring the assignment of global persistent identifiers for its items. Moreover, by exploiting a simple and effective file format supporting scheme it provides "bit preservation" of the digital content, while "functional preservation" will be provided only for the "supported" file formats. Finally, DSpace information's model is based on Communities, consisting of users and groups, containing Collections, which in turn contain items (the digital objects). Moreover, an adequate for most purposes workflow model is provided along with a simple administration toolkit. DSpace is a web-based system and all tasks are performed through web interfaces.

DSpace can be used "out of the box" for the generation of a digital repository in the case that content consists of independent digital documents. It provides simple, usable and effective resolutions of common problems, such as user and workflow management, persistence and indexing/searching. However, DSpace's aims do not include digitized content, inter-relation schemes and custom metadata in a fine-grained manner (per collection or sub-collection, for instance). Its customization capabilities also refer to this context and mainly are related to:

- user interface arrangements (such as logo placement, descriptions and color scheme)
- installation wide modification of the qualified DC metadata element set
- custom workflow steps setup

Although, DSpace provides several built-in facilities that simplify and speed-up the development of a digital repository, these features are highly coupled with each other and, mainly, coupled to the underlying database schema. For example, DSpace supports collections natively and the database schema reflects it by providing a distinct Collection table, holding collection related information. Enriching this information, by adding more table fields is possible, easy to be accomplished and straightforward. The same stands for analogous issues, such as metadata support, sub-collections or relations since all could be potentially supported by performing the necessary database schema modifications. The problem lies at two important issues arising by such modifications:

- Changes should be made to the "core" DSpace components. In order to perform significant modifications to its functionality, changes should be applied to both the database schema and relevant code.
- These changes, once made, break compatibility with future DSpace releases and the rest of DSpace installations, limiting the ability to benefit from future improvements, additions and extensions.

In simple cases, such as sub-collection support, the modifications or extensions required can be identified, designed based on current DSpace status and implemented in an adequately satisfying manner. In the case of more complex features, such as advanced collection management, in terms of heterogeneous collections support or different metadata set per collection, the required modifications may become extremely complicated. The point is whether the current architecture of DSpace will stand in the way for the development of such features, which were not included in its

initial design and development in the first place. Its database orientation indicates that this will be the case, practically deprecating its current form and design and requiring a re-implementation from the beginning.

4.3. Evaluation of supported features/System selection

The reasons for selecting Fedora to support University of Athens Digital Library are reported in the following: The digital object structure in Fedora provides the ability to support composite physical objects and preserve heterogeneous metadata of different categories (descriptive, technical, administrative or structural metadata). Furthermore, by using behaviors on each digital object it is manageable to dynamically implement collection-specific relations, in a unified and system-oriented fashion. It is also effective to support methods to create, edit or present digital objects, based on collection-specific templates. Collections are managed in the same way as other digital objects. Finally, Fedora's architecture provides the ability to extend the core system, by developing additional modules that communicate with it through the public APIs. It is a great advantage to customize the system without the need to intervene in the source code. In this occasion, the system can be easily upgraded when new versions become available.

5. Extending Fedora to support Folklore Collection

In order to support folklore collection extended features, some modeling customizations and system extensions should be considered. The proposed architecture is presented in figure1 and may host other collections with similar features as well. The main modules that should be attached over Fedora architecture are depicted in the figure. Two modules should be implemented by Libraries Computer Centre: object management and collection management module. An external module should also be used to provide indexing and searching features to an extended metadata set. These three modules access the Fedora APIs, so there is no need to communicate with the internal Fedora Repository System. They act as an intermediate level between the applications that will be developed, in order to provide administrator, cataloguers and user presentation services, and the Fedora system. All these applications belong in the presentation layer, the local modules and Fedora's APIs belong in the business logic layer and Fedora Repository System is in storage layer.

The main modeling conventions adopted for folklore collection special features are discussed in following. The main modules extending Fedora's functionality are also described.

Collection and sub-collection definition and description

Each data object used to represent a collection, contains a datastream for collection description and rights metadata and a datastream to define the data object's templates that will be created in the collection. All datastreams are implemented as inline XML

content. The main reason for defining data object templates in the collection is to provide guidelines in order to create and manage all collection data objects in a unified manner the. A specific template is used for each data object type. This template contains the definition of metadata fields for the data object and their characteristics (repeatable, mandatory, indexed, etc), the description of files that represent digital content, the proper relations and the behaviors associated to disseminate content and metadata.

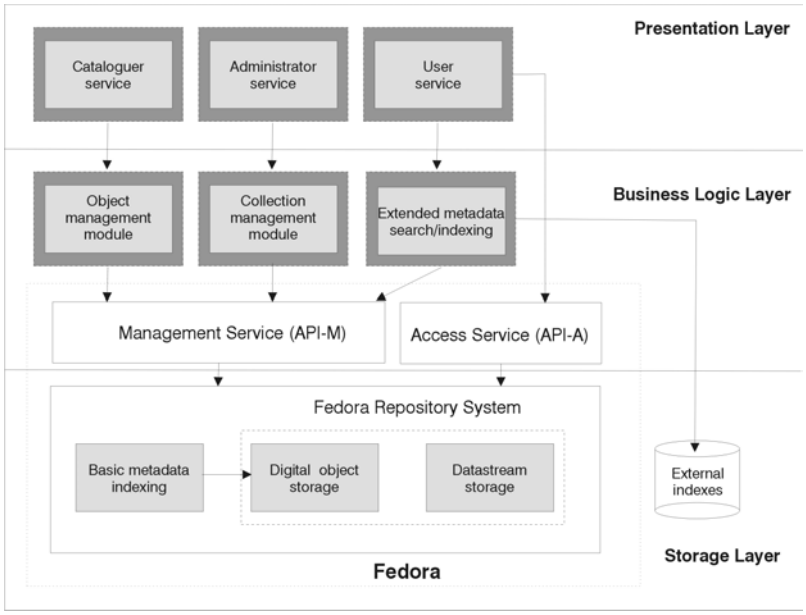

Figure 1: The proposed integrated DL architecture

An example of a digital object template from the photographs sub-collection is presented in figure 2. The template for a specific data object type (photo in the example) is enclosed in <*dobj*> tags. A template contains the following tags:

- <*field*> to define every field of the metadata set used by the data object type. When a data object is created, all these metadata fields are inserted in a datastream.
- <*file*> to define every file format necessary for the data object. Each file is associated with a datastream.
- <*disseminator*> to define the disseminators supported. A disseminator associates a behavior definition object (bdef) and a behavior mechanism object (bmech) with a specific datastream of data object.
- <relation> to define the permitted relations that the data object is able to perform. In the data object, the relations will be stored in a separate datastream.

An external java module manages collections using Fedora's management API (API-M). The main functionality of this module is to create collections and sub-collections, edit collection description metadata and import templates for the creation of data objects. When creating a sub-collection the description and rights metadata are copied from the parent to the child collection. The persistent id of the parent collection is

denoted on the content model type identifier (contentModelID) in data object's system metadata.

```
<dobj>
        <type>photo</type>
        <field>
                <name>dc:title</name>
                <indexed>true</indexed>
                <mandatory>true</mandatory>
                <repeatable>false</repeatable>
                <viewable>true</viewable>
                <label lang="eng">Photo title</label>
        </field>
        <field>
                <name>dc:subject</name>
                <indexed>true</indexed>
                <mandatory>false</mandatory>
                <repeatable>true</repeatable>
                <viewable>true</viewable>
                <label lang="eng">Subject</label>
        </field>
        <type>page</type>
        <file>
                <name>PHOTOHQ</name>
                <type>image/tiff</type>
                <label lang="eng">High quality photo</label>
        </file>
        <file>
                <name>THUMB</name>
                <type>image/jpeg</type>
                <label lang="eng">Photo thumbnail</label>
        </file>
        <relation>
                <name>page</name>
                <label lang="eng">Notebook page</label>
                <target_type>uoadl:10.page</ target_type >
        </relation>
        <disseminator>
                <name>goToPage</name>
                <label lang="eng">View notebook page</label>
                <datastream>STRUCT</ datastream>
                <bdef>uoadl:20</bdef>
                <bmech>uoadl:21</bmech>
        </disseminator>
</dobj>
```

Figure 2: Data object template for photographs sub-collection

Composite objects

The folklore notebook as a set of data objects that belong to three different types: main, chapter and page (figure 3). Every type corresponds to a data object template, which is defined on the appropriate collection. Type is defined in the c*ontentModelID* field of the digital object in the form *collection_pid.object_type (i.e. uoadl:10.chapter)* and it is pointing to the specific data object template. Main and chapter objects contain descriptive metadata specific to the notebook and the chapter respectively, together with structural metadata. Page objects contain the digital content (the page image in different formats) without descriptive metadata, together with structural metadata. Descriptive metadata, structural metadata and digital content are implemented in Fedora as separate datastreams of a data object.

The objects management module creates data objects of specific types based on the templates defined on the collection. The basic methods provided are: retrieve template guidelines, edit metadata, add files, create disseminators and relate data objects using the appropriate relations. All these actions are restricted from the guidelines given by the data object template.

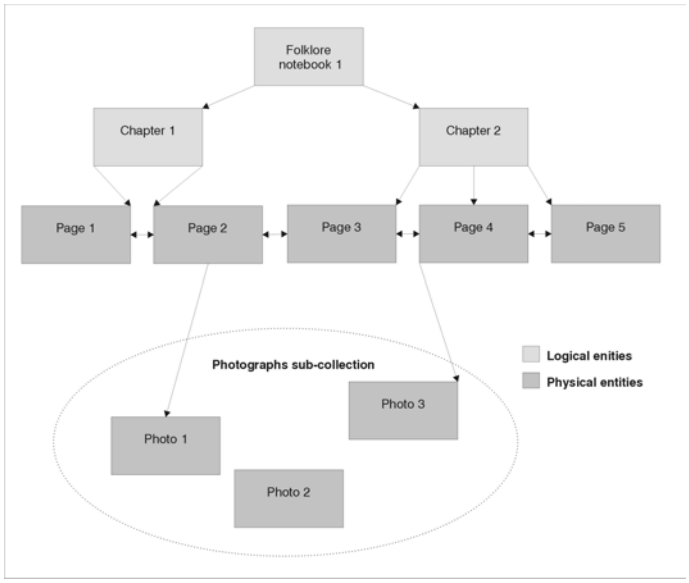

Figure3: Traveling Notebook Representation

Digital object relations

Relations are necessary to represent the structure of composite objects or to relate independent data objects belonging to the same or different collections. To implement relations, a special datastream is used on every data object. This datastream contains structure metadata, of the form:

```
<relation type= 'relation_type'>pid</relation>
```

The value *'relation_type'* denotes the type of the relation between the current data object and the one with the specified *pid*. The permitted relations for a data object are specified at collection level. The meaning of every relation is defined by the data object's disseminators. The relation types for the notebooks collection digital objects are: *'previous'* and '*next'* to navigate between pages, '*chapter'* and '*main'* to define current data object's chapter and main object, and '*photo'* to retrieve the photograph attached to the current page object. To extend navigation functionality in a notebook, a table of contents is used in each main digital object. This table is generated from the structural metadata of every data object connected to the specified notebook, by the 'main' relation. Table of contents is represented in main digital objects as an XML content datastream.

Although standard relation metadata can be used, such as Dublin Core relation element refinements (i.e. isParentOf, isReference, etc), a local metadata set was used, since it is more helpful to manage the structural metadata in a separate datastream, and more flexible to define specific relations depending on the collection needs, than using standard relation types. For example, the 'next' and 'previous' types have a special meaning in the notebook collection, helping the user to navigate between pages. In order to support interoperability with other digital repositories, we can use a mapping of these metadata to DC element refinements. In order to facilitate the defined relation in the presentation level, specific web services are implemented and

associated with behavior objects. Thus, end-user service communicates directly with Fedora Access API, in order to retrieve and use the appropriate behavior methods.

Extended metadata support
Every metadata model can be described in one or more datastreams of the digital object. Fedora supports indexing and searching methods, only for Dublin Core metadata element set ('DC' datastream) and digital object's system metadata. An external indexing application supports indexing and searching of other metadata sets on the XML files that store digital objects. Separate indexes must be created for every collection. Two open-source indexing applications that may be suitable for this purpose are Jakarta Lucene and Apache Xindice.

6. Conclusions

Athens University must support an integrated digital library framework for multiple heterogeneous digital collection development. The most important requirements imposed by collections were discussed in the paper. In order to select an open source digital library system for digital collection development, the folklore collection was chosen. A prototype implementation of folklore digital collection was performed using Fedora and DSpace. Based on the comparison results, it was decided to extend Fedora architecture to implement the integrated digital library platform supporting Athens University digital collection. While DSpace provides an enhanced 'out of the box' solution to develop collections including digital material, Fedora architecture is more expandable. This is the main reason for selecting it to implement folklore collection.

Reference

Consultative Committee for Space Data Systems (CCSDS), Reference Model for an Open Archival Information System (OAIS). Blue Book, Issue 1, January 2002.

Dublin Core Metadata Initiative: Dublin core Metadata Element Set, Version 1.1: Reference Description, available http://www.dublincore.org/documents/dces.

IEEE (2002), IEEE P1484.12.1/D6.4 Draft Standard for Learning Object Metadata, available http://ltsc.ieee.org/doc/wg12/LOM_WD6_4.pdf.

Open Society Institute, A Guide to Institutional Repository Software. 2nd Edition, January 2004.

Payette S. and Lagoze C. Flexible and Extensible Digital Object and Repository Architecture (FEDORA). ECDL '98, LNCS 1513, pp. 41-59, 1998.

Smith, M. et al., DSpace: An Open Source Dynamic Digital Repository. D-Lib Magazine, 9(1), January 2003.

Staples, T., Wayland, R. and Payette S., The Fedora Project: An Open-source Digital Object Repository Management System. D-Lib Magazine, 9(4), April 2003.

Yu, S.C., Lu, K.Y. and Chen, R.S. (2003) "Metadata management system: design and implementation", The Electronic Library, Vol. 21, No 2 pp. 154-164.