A digital library service for the small

Stavros Angelis

Panos Constantopoulos

Dimitris Gavrilis

Christos Papatheodorou

Digital Curation Unit, “Athena” Research Centre

Artemidos 6 & Epidavrou, Maroussi
Athens, Greece
+302106875365

{ s.angelis, p.constantopoulos, d.gavrilis, c.papatheodorou, }@dcu.gr

ABSTRACT

In this paper, we present MOPSEUS, a lightweight digital library service based on the Fedora system. This service was created to address the needs of small libraries without support from technical staff. Hence, MOPSEUS attempts to balance flexibility against ease of installation, configuration and use. The service is available as a standard Java Web servlet, uses no external databases or other systems and can easily be deployed on top of any Fedora installation. Preliminary tests concerning the ease of installation and use are encouraging. We contend that facilitating the introduction of digital library infrastructures in the small may contribute to spreading digital curation practices.

Categories and Subject Descriptors

D.2.11 [**Software Architectures**]: Domain-specific architectures,

General Terms

Management, Documentation, Design, Experimentation, Human Factors, Standardization.

Keywords

Digital libraries, metadata schema, Fedora digital library, open architecture, preservation.

1. MOTIVATION

In the last decade, the library community, supported by the advances in information and communication technologies, demonstrated a rapid growth characterized by its entry into to the digital world. This move goes along with the emergence of a large number of complex digital library systems (both commercial and open source) that can handle millions of records and can serve large numbers of users with various characteristics and needs. However, the target group of such systems consists mostly of medium to large size libraries with sufficient budget and technical staff to acquire and maintain them. If the library community were sorted by size, the following categories would emerge:

- *Large size libraries:* usually national libraries or academic libraries.
- *Medium size libraries:* usually found in companies, small universities, branches of large academic libraries or technical schools.
- *Small size libraries:* usually found in schools and less populated areas, such as small towns and villages.

These three categories have different characteristics (such as budget, number of employees, and number of users) that can be seen in Table 1 below. The budget, employee and user numbers increase with library size, whereas the category population grows inversely to the library size.

Table 1. Characteristics of libraries (per size)

	Large size	**Medium size**	**Small size**
Number	small	small - medium	large
Budget	large	medium	small - very small
Number of employees	large	medium - small	small - very small
Number of users	large - very large	medium	small - very small

In Greece, for example, there are a few tens of large libraries (mostly academic) and a few hundred small size ones (mostly found in small islands or schools). Similar examples can be found all over the world [4]. The financial and technical requirements of a modern digital library system exclude a large community of libraries, namely the small ones [5]. The small libraries community usually has inadequate funds and staff to use such systems, even if these are free and open source [6].

The small library community cannot afford expensive commercial library systems, but also faces great difficulties in setting up and maintaining most open source systems because of lack of relevant expertise [7]. Small libraries usually have no more than two employees, who are often volunteers and in most cases non-technical. Therefore, we set forth the following set of requirements for a digital library service suitable for employment in small libraries:

- Ease of installation: someone without technical background should be able to install the digital library quickly and easily in a few steps.
- Ease of configuration: the librarian should be able to configure the system quickly and easily.
- Flexibility: the system must be able to support different, custom metadata schemata. This would enable the service to adapt to different library needs (e.g. school, museum library, public library, etc.).
- Web-based: users should be able to access the system remotely through the Internet and through a familiar overall interface. In addition, it should be possible to setup the system on a single server (e.g. a small library server/workstation) and provide access to more than one librarians/users.
- Interoperability: the system must be compatible with other systems and able to expose its collection to external agents (e.g. PMH harvesters).
- Preservation and backup: the system must be able to handle data preservation and allow the user to easily backup the database.

This paper presents MOPSEUS, a digital library service based on the idea to address the needs of this community for ease of installation, configuration and use. Furthermore, the initiative to facilitate the introduction of digital library infrastructures in the small libraries, may contribute to spreading digital curation practices. The availability of a simple and convenient tool like MOPSEUS to small library communities enables them to perform simple, yet crucial operations required for digital curation, such as the ingestion, modification and processing of digital objects, while enabling the specification of useful metadata, and providing a simple backup process. Furthermore, ongoing work on MOPSEUS focuses on providing mechanisms for the transformation and interoperability between different metadata schemata. The aforementioned features of MOPSEUS are still a long way from satisfying all the requirements for digital curation, as presented in the lifecycle model of the UK's Digital Curation Centre (DCC) [9] or the more recent DCC&U lifecycle model [10] of the Digital Curation Unit (DCU). However, MOPSEUS still provides an initial subset of necessary operations for digital curation. These features may act as a stepping stone, making it far easier, thus acceptable, for small library communities to consider satisfying additional curation operations, as specified in [9,10].

2. SYSTEM ARCHITECTURE

In order to address the above requirements, two widely recognized open source systems were considered: DSpace [3] and Fedora [1]. The first, though generally adopted by the majority of the academic library community, has a more complex installation and configuration process. Furthermore, its metadata schema and preservation architecture do not meet the flexibility requirement. On the other hand, the Fedora system has an open, service oriented architecture and an easy deployment process, thus better meeting most of the requirements:

- Interoperability: Fedora's service oriented architecture and embedded OAI provider greatly enhance its interoperability features [2].
- Preservation & backup: Fedora stores its digital objects in the file system and a simple backup process is reduced to just copying a specific folder to some external media.
- Flexibility: Fedora's object architecture involves multiple datastreams and an interconnection capability through the Fedora ontology, thus making Fedora extremely flexible.

The main drawback of Fedora, on the other hand, is its lack of a user-friendly interface.

2.1 MOPSEUS

MOPSEUS is a Web based system built on top of Fedora and consists of three main subsystems (Figure 1): communication, datastream management, user functions. The communication subsystem employs the SOAP and REST protocols in order to communicate with the Fedora server.

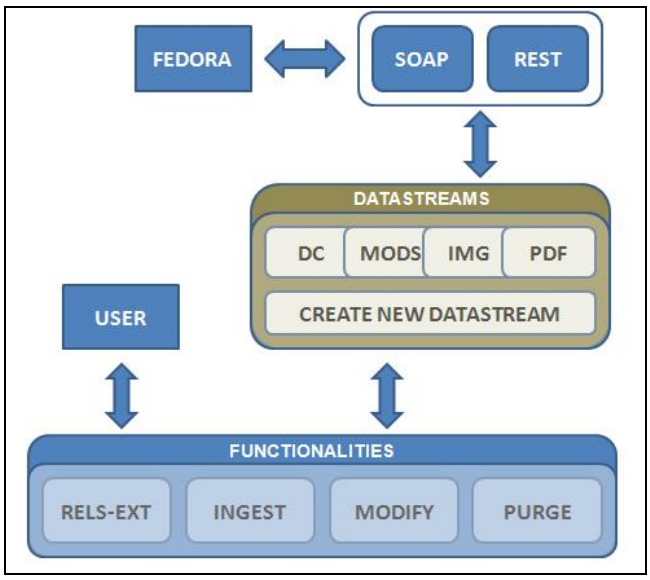

Figure 1. The MOPSEUS architecture

The user interacts with the Fedora system through a set of basic functions that include: *object ingest*, *modify*, *purge*, *object relations*. These functions refer to Fedora digital objects that contain datastreams. These datastreams are handled by a separate subsystem that is responsible for rendering the datastream contents to the user through a friendly interface. Furthermore, the user can extend the system by adding new datastreams.

2.2 Metadata schema

In contrast to most Fedora-based systems, MOPSEUS does not require an external database management system (such as MySQL or Oracle), but rather uses Fedora's internal McKoi database system instead. This approach greatly simplifies the installation process since the user only has to install Fedora (a very simple process) and then simply copy the MOPSEUS `.war` file to the Fedora installation. The problem with not having an external database system is where to store all the configuration data that a digital library needs. The solution is to encode the configuration data as digital objects in the repository itself, thus

creating a self-describing repository. A separate namespace (admin) was created to hold these data.

The metadata schemata available in MOPSEUS are described in XML and stored as datastreams in a specific digital object. For each entry in the metadata schema list a separate datastream must be created in the digital object in order to describe that schema. Only the Dublin Core metadata schema is mandatory, a choice intended to ensure interoperability. For example, if the user wants to extend the existing metadata schemata by including a new one (e.g. MODS), he must add an entry to the list of the available metadata (e.g. with ID=MODS) schemata and then create a new datastream with label MODS that describes the MODS metadata schema using XML. An example of this datastream can be seen in Figure 4. MOPSEUS will translate this XML based schema into user-friendly HTML forms.

```
<schema id=”MODS” name=”MODS Schema”>

<element id=”title” name=”title” type=”text”>
<format attribute=”false” attribute_name=”” write=”false” write_id=”true”></format>
<style css=”normal_text” size=”40” type=”text”>
<display>Title</display>
<description>Enter the title of the article</description>
</element>

<element id=”type” name=”type” type=”list”>
<format attribute=”true” attribute_name=”type” write=”true” write_id=”true”></format>
<style css=”list_text” size=”40” type=”select”>
<display>Publication Type</display>
<description>Enter the publication type of the article</description>
<value>Journal</value>
<value>Conference</value>
<value>Book</value>
</element>

</schema>
```

Figure 2. An example of a metadata schema description

In Figure 2, we give an example of a metadata schema description in XML, namely a sample MODS metadata schema containing two elements, **title** and **type**. The user may define an arbitrary number of elements and nest them in every possible way. For each element, the user has to provide the following information:

- Basic information: element’s id, name and type. Type can have the following values: label, text, list. *Label* is used to group elements together. *Text* is used to present the user with various text fields. *List* provides the user with select lists.
- Format information: the format information describes how the element data will be encoded when creating the XML datastream.
- Style information: the style information enables the user to choose the style class that will be used for rendering the element in html along with its size (e.g. in the case of text area box: rows and columns).
- Display and description information: the display and description information defines the label and a short description of its function.

2.3 Web based interface

In order to overcome Fedora’s lack of a user friendly interface, a Web-based interface was built on MOPSEUS. The embedded tomcat Web server that came with Fedora made deployment even easier since the user only needs to copy the MOPSEUS`.war` file to a specific directory of the tomcat server.

Basic functions supported include *search*, *ingest*, *modify* and *purge*.

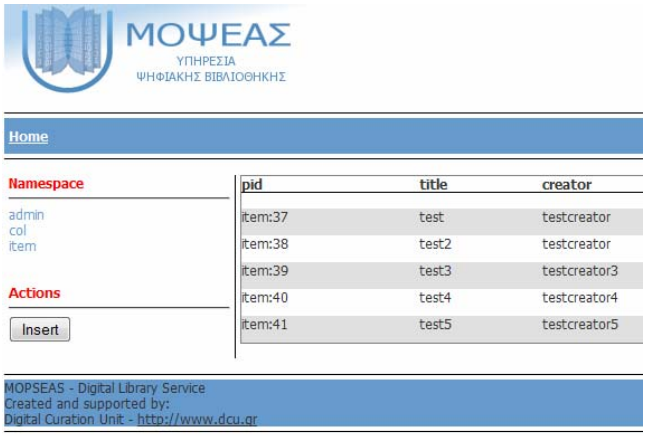

Figure 3. The basic search screen

The main page of MOPSEUS (SEUS (see Figure 3), displays the digital objects contained in the library, allows the user to ingest an object and perform basic search functions. These functionalities are provided by a user-friendly web-based interface, minimizing the technical skills required by the cataloguer.

By clicking on an object, the user is presented with its Dublin Core record and a list of the other available datastreams of that object (Figure 4). The user can also insert a new datastream, or purge an existing one. The fact that most of the functions of MOPSEUS are handled from these two pages minimizes the learning overhead.

3. PRELIMINARY EVALUATION

A preliminary experimental investigation of the MOPSEUS system usability has been performed using an expert – based methodology [8]. In particular the Cognitive Walkthrough method was selected which concentrates mainly on the difficulties users may experience in learning to operate an application to perform a given task. During a Cognitive Walkthrough process the expert

first determines the exact sequence of correct task performance, and then estimates, on a screen by screen basis, the likely success or failure of the user in performing such a sequence. The main characteristic of this approach is that the expert must make an informed guess of the likely reaction of users and explain why certain interface attributes are likely to cause users difficulties.

In our experiment ten librarians with significant expertise in the development of digital collections and the usage of digital library technologies were asked to install the system, and use it by ingesting and retrieving digital objects and metadata. After a 5-days usage period the participants were asked to fill a short questionnaire, expressing their attitude on a 5-point Likert scale towards a set of criteria including learnability, ease of use, navigation, aesthetic appearance, use of established terminology and finally the perceived usefulness of the system's functions (i.e. their perception of the improvement of their productivity on digital collections management).

To better assess whether the system fits and supports adequately their everyday tasks, a part of the questionnaire focused on

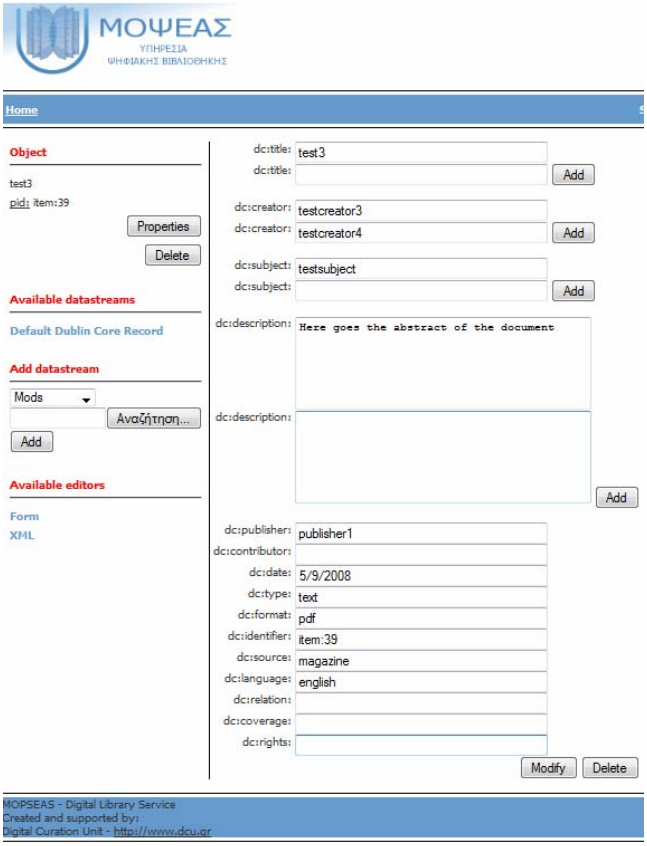

Figure 4. The object details screen

particular functions, such as searching and browsing, ingesting and deposing and metadata editing.

The results were quite encouraging since the overall satisfaction from the system rated an average grade of 4.4 in the 5-points Likert scale. Table 2 presents the average values on the criteria used in the evaluation experiment. Although the users rated highly the searching, browsing and depositing functionalities of MOPSEUS, they requested a simpler installation procedure and the incorporation of help and instructions in each page.

Table 2. Evalution criteria average values

Criterion	Value
Learnability	4.2
Aesthetic appearance	3.8
Navigation	4.4
Terminology	4.6
Ease of use	4.6
Usefulness	3.8
Usability	4.4
Overall Satisfaction	4.4

Furthermore, some users requested an advanced search facility and the ability to create collections and add items to them. The lack of these functionalities is responsible for the low usefulness average. Regarding the collection definition and development functionality, it should be mentioned that the users were not familiar with the Fedora relationships that allow the user to create collections of items. Regarding the installation process, a simpler installation procedure was created in response that helped users perform the installation quicker and easier. Finally, it should also be noted that the aesthetics evaluation which rated relatively low, had a large standard deviation (about 1.5) while the other evaluation criteria, had a standard deviation of approximately 0.5.

4. CONCLUSIONS AND FUTURE WORK

MOPSEUS is a lightweight, open SOA digital library service based on Fedora, easy to install and maintain, provided as open source by the Digital Curation Unit. These features make MOPSEUS suitable for small libraries with no specialized technical staff. At this early stage MOPSEUS has shown some promising results. Several more features need to be implemented, such as workflow management, advanced search and easier parameter configuration.

5. ACKNOWLEDGMENTS

We would like to thank Prof. Antonios Deligiannakis for reviewing this paper.

6. REFERENCES

[1] Payette, S., Lagoze, C., 1998. Flexible and Extensible Digital Object and Repository Architecture (FEDORA). Second European Conference on Research and Advanced Technology for Digital Libraries, Heraklion, Crete, Greece September 21-23, 1998, Springer, 1998, (Lecture notes in computer science; Vol. 1513). http://www.cs.cornell.edu/payette/papers/ECDL98/FEDORA.html

[2] Staples, Thornton, Ross Wayland and Sandra Payette, 2003. The Fedora Project: An Open-source Digital Object Repository System. D-LIb Magazine, April 2003. http://www.dlib.org/dlib/april03/staples/04staples.html

[3] DSpace repository: http://www.dspace.org

[4] The Libraries in Small Schools Project. http://smallschools.ischool.washington.edu/

[5] Roberts,Gary, 2005. Small Libraries, Big Technology. Computers in Libraries 25.3 (March 2005): 24- . Academic Search Premier. EBSCO on LINCCWeb. Central Florida Community Coll. Lib., Ocala. 8 June 2005. http://www.linccweb.org

[6] Calhoun, K., Hello and On Small Libraries, http://community.oclc.org/metalogue/archives/2008/06/hello-and-on-small-libraries.html

[7] Fuller, D., School Library Journal & San Jose State University 2006 Automation Survey: The systems are changing. But school libraries aren't. School Library Journal 10-01-2006 http://www.schoollibraryjournal.com/article/CA6376081.html?industryid=47074&q=small+library+technology

[8] Nielsen, J., Usability engineering. Morgan Kaufmann,Boston, USA (1993).

[9] Higgins Sarah, 2007. The dcc curation lifecycle model. Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries. Pittsburgh PA, PA, USA, pp. 453-453.

[10] Constantopoulos, P., Dallas, C., Androutsopoulos, I., Angelis, S., Deligiannakis, A., Gavrilis, D., Kotidis, Y., Papatheodorou, C., 2008. DCC&U: An Extended Digital Curation Lifecycle Model. Proceedings of the 4th International Digital Curation Conference. Edinburgh, Scotland. December 2008.