Comparing Open Source Digital Library Software

George Pirounakis
Libraries Computer Centre, University of Athens
Panepistimiopolis, Ilisia, 15784
Athens, GREECE
elourdi@lib.uoa.gr

Mara Nikolaidou
Harokopio University of Athens
70 El. Venizelou St, Kallithea, 17671
Athens, GREECE
mara@di.uoa.gr

Topic: Digital Libraries: Design and Development, Open Source Digital Libraries, Evaluation

Keywords: Digital Library Software, Digital Library Architecture, Feroda, DSpace, GreenStone, KStone, EPrints

1. Introduction

The last years a great number of digital library and digital repository systems have been developed by individual organizations -mostly Universities- and given to the public as open-source software. The advantage of having many choices becomes a great headache when selecting a Digital Library (DL) system for a specific organization. To make the decision easier, we compared five such systems that are publicly available using an open source license, are compliant with Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [1] and already have a number of installations worldwide. Using these basic restrictions we selected for comparison the following five (5) broadly used DL systems:

- DSpace [2], developed by the MIT Libraries and Hewlett-Packard Labs (BSD open source license)
- Fedora [3], jointly developed by Cornell University and the University of Virginia Library (Educational Community License)
- Greenstone [4], produced by the University of Waikato (GNU General Public License)
- Keystone [5], developed by Index Data (GNU General Public License)
- EPrints [6], developed by the University of Southampton

Each of these systems was been thoroughly studied based on basic characteristics and system features described in the following sections. The latest versions of those systems were examined. On February 2007 (when writing this article) the versions provided were: DSpace 1.4, Fedora 2.2, Greenstone 3, Keystone 1.5 and EPrints 3. The DL systems are compared based on stated characteristics and the level of support on each of them. In section 2, the characteristics needed by a modern DL system are discussed. In section 3, the five DL systems are compared based on each of the DL characteristics and the results are summarized in a score table. Finally, in section 4, the results of this comparison are commented and cases for which, each of these systems is suitable, are proposed.

2. DL Systems Characteristics

The basic characteristics and features that expect from a modern integrated DL software are:

1. *Object model*. The internal structure of the digital object [7] (entity that integrates metadata and digital content) in the DL system. Existence of unique identifiers for the digital object and every part of it is also important to ensure preservation and easy access.

2. *Collections and relations support*. Collection description metadata, definition of sub-collections and templates that describe the format of the digital objects or the presentation of the collection. Definition of relations between objects of the same or different types.
3. *Metadata and digital content storage*. The storage capabilities are stated, along with the preservation issues. It is important for the DL system to ensure standard as long as user defined metadata sets and multiple formats of digital content.
4. *Search and browse*. The mechanisms used for indexing and searching of the metadata. It is important for the DL system to support indexing not only for a restricted metadata set, but also for selected metadata fields.
5. *Object management*. Methods and user interfaces provided from the DL system to manipulate (insert, update and delete) metadata and digital content.
6. *User interfaces*. Provided user interfaces for end-user access on the DL, its collections and the digital objects.
7. *Access control*. Support for users and groups, authentication and authorization methods. Level of restriction for access and update (DL, collection, digital object and content).
8. *Multiple languages support*. Multiple languages should be supported in the user interface, in the metadata fields and in the digital content. The character encoding is of great importance in order for the DL systems to be fully multilingual.
9. *Interoperability features*. Standards that the DL systems support in order to ensure interoperability with other systems. Export of the digital objects in open standard formats is also important.
10. *Level of customization*. Customization of the DL system in collection level, the format of the digital objects and the services provided. The quality and methods provided by the application programming interfaces (APIs) of the DL systems.

3. DL Systems Comparison

In the following, the five open access DL systems are compared based on the characteristics identified in the previous section. The level of support of each characteristic and specific considerations for each DL system are discussed.

Object model

Dspace: The basic entity in DSpace is *item*, which contains both metadata and digital content. Qualified Dublin Core (DC) [8] metadata fields are stored in the item, while other metadata sets and digital content are defined as bitstreams and categorized as bundles of the item. The internal structure of an item is expressed by structural metadata, which define the relationships between the constituent parts of an item. DSpace uses globally unique identifiers for items based on CNRI Handle System. Persistent identifiers are also used for the bitstreams of every item.

Fedora: The basic entity in Fedora is *digital object*. The internal structure of digital object is determined from the Fedora Object XML (FOXML), which is based on Metadata Encoding and Transmission Standard (METS) [9]. Digital object contains metadata and digital content (both treated as datastreams). Digital object also contains links to the behaviors defined for it. A unique persistent identifier is used for every digital object. Datastreams are also uniquely identified by a combination of the object persistent identifier and the datastream identifier.

Greenstone: Basic entity in Greenstone is *document,* which is expressed in XML format. Documents are linked with one or more resources that represent the digital content of the object. Each document contains a unique document identifier but there is no support for persistent identifiers of the resources.

Keystone: Basic entity in Keystone is *document*. The internal structure of each document is defined in a user defined XML Schema corresponding to a specific document type. The directory structure of the documents represents the object’s structure. A persistent identifier is not used to uniquely identify documents.

EPrints: Basic entity in EPrints is the *data object*, which is a record containing metadata. One or more documents (files) can be linked with the data object. Each data object has a unique identifier.

Collections and relations support

Dspace: Supports collections of items and communities that hold one or more collections. An item belongs to one or more collections, but has only one owner collection. It is feasible to define default values for the metadata fields in a collection. The descriptive metadata defined for a collection are the title and description. There is no support of relations between different items.

Fedora: Fedora supports collections using RELS-EXT datastream that contains a basic relationship ontology. In this datastream the relationships between digital objects (like isMemberOfCollection or isPartOf) are expressed using RDF. Fedora does not provide a mechanism to manipulate these relations.

Greenstone: A collection in Greenstone defines a set of characteristics that describe its functionality. These characteristics are: indexing, searching and browsing capabilities, file formats, conversion plugins and entry points for the digital content import. There are also some characteristics for the presentation of the collection. The representation of hierarchical structure in text documents is supported for chapters, sections and paragraphs. The definition of specific sections in text document is implemented through special XML tags. XLinks in a document can be used to relate it with other documents or resources.

Keystone: Collections in Keystone are not defined as entities but they are imposed by the directory structure of the documents. The document XML Schema specifies common behavior (elements are viewable, repeatable, mandatory, multi-lingual, use a restricted vocabulary) for the documents of the specific type. There is no definition of relations between documents, except using URLs in specific metadata fields.

EPrints: There is no consideration of collections in EPrints. Data objects are grouped depending on specific fields (subject, year, title, etc). There is no definition of relations between documents, except using URLs in specific metadata fields.

Metadata and digital content storage

Dspace: Dspace stores qualified DC metadata in a relational database (PostgreSQL or Oracle). Other metadata sets and digital content are represented as bitstreams and are stored on filesystem. Each bitstream is associated with a specific bistream format. A support level is defined for every bistream format, indicating the level of preservation for the specified file format.

Fedora: Metadata and digital content are both considered datastreams of the digital object. Datastreams can be stored (a) internally on the digital object XML file, (b) on filesystem as managed content or (c) on an external source. One or more metadata sets can be concurrently used, while different file formats can be stored as separate datastreams in a digital object. Basic technical metadata are stored for each datastream like MIME type, file size and checksums, ensuring content preservation. Fedora supports versioning of specified datastreams, allowing user to access older datastream instances.

Greenstone: Both documents and resources are stored on filesystem. Metadata are user defined and are stored in documents using an internal XML format.

Keystone: Each object in Keystone contains its metadata in an XML document. The metadata are not restricted to a specific metadata standard but are stated in a user defined XML Schema denoting each document type. Digital content is stored in the directory structure that contains the XML documents.

EPrints: Metadata fields in EPrints are user-defined. The data object, containing metadata, is stored in a MySQL database and the documents (digital content) are stored on filesystem.

Search and browse

Dspace: Provides indexing for the basic metadata set (qualified DC) by default, using the relational database. Indexing of other defined metadata sets is also provided using Jakarta Lucene API. Lucene supports fielded search, stemming and stop words removal. Searching can be constrained in a collection or community. Also, browsing is offered by default on title, author and date fields.

Fedora: Default indexing is provided for the DC metadata set and digital object's system metadata (persistent identifier, creation/modification date, label, content model). Indexing and searching is managed from a relational database (MySQL, Oracle or PostgreSQL). Searching is available in all indexed fields using constraints on a combination of fields. A generic search (gSearch) is also provided using Lucene or Zebra search engines. In addition, relationships between digital objects are indexed and are searchable using the Fedora Resource Index. Browsing mechanism is not provided.

Greenstone: Indexing is offered for the text documents and specific metadata fields. Searching capabilities provided for defined sections in a document (Title, chapter, paragraph) or in whole document. Stemming and case sensitive searching is also available. Managing Gigabytes (MG) open-source applications is used to support indexing and searching. Browsing catalogs can be defined for specific fields using hierarchical structure.

Keystone: Indexing is supported on specified document types for the whole metadata set. Free text searching is offered. Browsing mechanism is not provided.

EPrints: Indexing is supported for every metadata field, using the MySQL database. Full text indexing is supported for selected fields. Combined fielded search and free text search are provided to the end-user. Browsing is provided using specified fields (e.g. title, author, subject).

Object management

DSpace: Items in DSpace are created using the web submission user interface or the batch item importer, which ingests XML metadata documents and the constituent content files. In both cases a workflow process may initiate depending on the collection configuration. The workflow can be configured to contain from one to three steps where different users or groups may intervene to the item submission. Collections and communities are created using the web user interface.

Fedora: Creation of digital objects is feasible using the Administrator client or the batch import utility (XML files in METS or FOXML format). Metadata addition or editing is provided through a text editor in Administrator client. The same client is used for addition and removal of digital content (as datastreams).

Greenstone: New collections and the contained documents are built using the Greenstone Librarian Interface or the command line building program.

Keystone: The content management system of Keystone provides the web interface for editing documents. It allows specified users to manage the content of documents as long as the files structure.

EPrints: A default web user interface is provided for the creation and editing of objects. Authority records can be used helping the completion of specific fields (e.g. authors, title). Objects can also be imported from text files using multiple formats (METS, DC, MODS, BibTeX, EndNote).

User interfaces

DSpace: A default web user interface is provided in order for the end-user to browse a collection, view the qualified DC metadata of an item and navigate to its bistreams. Navigation into an item is supported through the structural metadata that may determine the ordering of complex content (like book pages or web pages). A searching interface is provided by default that allows the user to search using keywords.

Fedora: The web interface of Fedora provides a search environment to the end-user, where he/she may execute simple keyword or field search queries. The default view of digital objects is restricted to the presentation of the system metadata and the datastreams.

Behavior digital objects define the presentation or manipulation methods of datastreams. A developer may build specific web services and attach them on digital objects as behaviors. A DC metadata viewing page and an image manipulation applet are provided as default behaviors.

Greenstone: The default web user interface provides browsing and searching into collections, navigating into hierarchical objects (like books) using table of contents. Presentation of documents or search results may differ depending on specified XSLTs.

Keystone: Presentation of a document is controlled by an XSLT stylesheet that reflects the associated document type. The main web user interface is based on a portal like environment. In this environment a user may browse the documents directory structure and search in the digital library.

EPrints: The web user interface provides browsing by selected metadata fields (usually subject, title or date). Browsing can be hierarchical for subject fields. Searching environment allows user to restrict the search query using multiple fields and select values from lists.

Access control

DSpace: It supports users (e-people) and groups that hold different rights. Authentication is provided through user passwords, X509 certificates or LDAP. Access control rights are kept for each item and define the actions that a user is able to perform. These actions are: read/write the bitstreams of an item, add/remove the bundles of an item, read/write an item, add/remove an item in a collection. Rights are based in a default-deny policy.

Fedora: It supports users and groups authorized for accessing specific digital objects using XACML policies. Authentication is provided through LDAP or for specific IP addresses.

Greenstone: A user in Greenstone belongs to one of two predefined user groups: an administrator or a collection builder. The first user group has the right to create and delete users, while the second builds and updates collections. End-users have access to all the collections and the documents.

Keystone: A simple access control is supported where you can define administrators and simple users that have access rights on specific parts of the documents structure.

EPrints: Registered users in EPrints are able to create and edit objects. Users are logged in using their username and password pair.

Multiple languages support

All the DL systems use Unicode character encoding, so the support of different languages can be supported. Every system can use multiple languages in the metadata fields and digital content. Keystone and EPrints provide an XML attribute on metadata fields to define the language used for the field value. Greenstone provides ready to use multilingual interfaces already translated in many languages.

Interoperability features

All the DL systems support OAI-PMH in order to share the metadata of the DL with other repositories. Greenstone and Keystone also support Z39.50 protocol for answering queries on specific metadata sets. Fedora and DSpace are able to export digital objects as METS XML files. Both systems also use persistence URIs to access the digital content providing a unified access mechanism to external services. DSpace also supports OpenURL protocol providing links for every item page. EPrints exports data objects in METS and MPEG-21 Digital Item Declaration Language (DIDL) format.

Level of customization

Dspace: Although DSpace has a flexible object model is not so open in constructing very different objects with independent metadata sets because of its database oriented architecture. The user interface is fixed and provides only minor presentation interventions. Another disadvantage is the full support of only specific file formats as digital content.

Fedora: In Fedora every digital object can follow a different content model that describes its format. It is also possible to provide multiple behaviors in it that determine the access and manipulation methods of the digital object. These two characteristics result in a fully customizable DL. The user interface, although by default is poor, is fully customizable based on two APIs (Access API and Management API).

Greenstone: It provides customization for the presentation of a collection based on XSLTs and agents that control specific actions of the DL. Greenstone architecture provides (i) a back end that contains the collections and the documents as long as services to manage them and (ii) a web based front end that is responsible for the presentation of collections, documents and their searching environment.

Keystone: Document’s structure is based on a customized document type, which is formed by an XML Schema. In addition the presentation of a document is dependent on the XSLTs associated with the document type. The separation of document storage and presentation layer, as long as the typing of documents provides a fully customizable DL architecture.

EPrints: The data objects in EPrints contain user defined metadata. Plug-ins can be written in order to export the data objects in different text formats. A Core API in Perl is provided for developers who prefer to access basic DL functionality.

Based on the above analysis, the five DL systems were graded for each of the characteristics. The minimum score is 1 and the maximum is 5.

Characteristics	***DSpace***	***Fedora***	***Greenstone***	***Keystone***	***EPrints***
Object model	4	5	3	3	2
Collection support and relations	4	4	5	2	1
Metadata and digital content storage	4	5	3	3	3
Search and browse	4	3	4	2	4
Object management	4	2	2	3	4
User interfaces	4	2	4	4	4
Access control	5	4	2	2	2
Multiple languages support	3	3	4	4	4
Interoperability features	5	5	4	4	5
Level of customization	3	5	4	5	3

4. Conclusion and Suggestions

It is difficult to propose one specific DL system as the most suitable for all cases. Each system has its advantages and drawbacks, as stated in the above comparison, categorized by basic DL system characteristics and features. That comparison can only be used as a guideline by an organization in order to decide if one of these DL systems is suitable to host its digital collections. Usually the needs for each organization vary depending on the number of collections, the types of objects, the nature of the material, the frequency of update, the distribution of content and the time limits for the development of a DL. In the next paragraphs, guidelines for the selection of a DL system are provided depending on different organization needs.

1. Consider a case where an institution or university needs a digital repository for research papers and dissertations produced by students and stuff. In that case, the most appropriate DL system is DSpace, since it by default represents communities (e.g. university departments) and collections (e.g. papers and dissertations), while workflow management supported is important for item submission by individuals.

2. Consider a case where an organization needs one digital collection to publish its digital content in a simple form, in strict time limits. In addition, the organization prefers to integrate the web interfaces of the DL with a portal like website. In that case the most appropriate DL systems are Keystone or EPrints, since they separate the concerns of presentation and storage, are not bind to specific metadata standards and provide simple web interfaces for the submission and presentation of documents and metadata.

3. Consider a case where an organization is responsible to digitize collections from libraries, archives and museums and host them in a single DL system. The organization has human resources and the amount of time in order to customize the DL system and develop extra modules. The highest priority needs are the support of preservation issues, the use of multiple metadata standards and the different formats of digital content. In that case the most suitable DL system is Fedora, since it provides a very customizable modular architecture. Although it does not provide easy to use web interfaces or built-in functionality, it is the best choice for the case where many collections and different material must be hosted.

4. Consider a case where an organization wants to electronically publish books in an easy to use customizable DL system. In that case the most appropriate DL system is Greenstone, since it is easy to represent books in a hierarchical manner, using table of contents, while the full text of chapters can be searchable.

References

1. C. Lagoze and H. Van de Sompel. The Open Archives Initiative: Building a low-barrier interoperability framework. In Proceedings of the Joint Conference on Digital Libraries (JCDL ’01), 2001.
2. DSpace Federation. Available at http://www.dspace.org/
3. Fedora Project. Available at http://www.fedora.info/
4. Greenstone Digital Library Software. Available at http://www.greenstone.org/
5. Keystone DLS. Available at http://www.indexdata.dk/keystone/
6. EPrints for Digital Repositories. Available at http://www.eprints.org/
7. R. Kahn and R. Wilensky. A Framework for Distributed Digital Object Services. Corporation of National Research Initiative - Reston USA, 1995. Available at http://www.cnri.reston.va.us/k-w.html
8. DCMI Metadata Terms. Dublin Core Metadata Initiative. Available at http://www.dublincore.org/documents/dcmi-terms/
9. METS: An Overview & Tutorial. Library of Congress. Available at http://www.loc.gov/standards/mets/METSOverview.v2.html