Building Digital Collections using Open Source Digital Repository Software: A Comparative Study

George Pyrounakis
Dept. of Informatics and Telecommunications,
University of Athens
Greece

Mara Nikolaidou
Dept. of Informatics and Telematics,
Harokopio University of Athens
Greece

Michael Hatzopoulos
Dept. of Informatics and Telecommunications,
University of Athens
Greece

ABSTRACT

The last decade a great number of digital library and digital repository systems have been developed and published as open-source software. The variety of available software systems is a factor of confusion when an organization is planning to build a repository infrastructure to host its collections. To simplify the decision process five widely used open-source repository software systems are compared, namely DSpace, Fedora, Greenstone, EPrints and Invenio. In addition to the comparison of these software systems and their characteristics' description, we propose the most suitable systems for different cases of digital collections. Using five collection paradigms that represent case studies of different content and functionality, an organization can be directed to select a repository software matching its criteria.

Keywords: Repository, Digital Library, Digital Collection, Open-Source Software, Fedora, DSpace, Greenstone, EPrints, Invenio

INTRODUCTION

The last decade a great number of Digital Library (DL) and Digital Repository (DR) systems have been developed and published as open-source software. The variety of available software systems becomes a headache when an organization plans to build a repository infrastructure to host its collections. Fortunately, there are many articles and surveys that evaluate or compare open-source DR and DL software. One of the first guides for selecting open-source repository software, based on the features and benefits of 9 different repositories, is provided by the Open Society Institute (2004). An extensive checklist for evaluating DL software is drafted by Goh et al. (2006). Also, two recent papers compare and evaluate some current open-source DR and DL software (Masrek & Hakimjavadi, 2012; Tramboo et al., 2012). The main scope of these papers is the comparison of the software systems based on some quantitive and quality characteristics, in order for interested organizations to select the proper system for their digital collections. In our study we try to go a step further and in addition to the comparison of DR software systems and their characteristics' description, we propose the most suitable systems for different collection types. Using five collection paradigms that represent case studies of different content and functionality, an organization can be directed to select a repository software matching its criteria.

We used the following three restrictions in order to select the repository software systems that participate in the comparative study. The repository systems:

1. are publicly available using an open-source license,
2. are compliant with the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) (Lagoze & Sompel, 2001),
3. have a large number of installations worldwide.

Software	**Initially developed by**	**License**	**Website**
DSpace	MIT Libraries and Hewlett-Packard Labs	BSD open source license	http://www.dspace.org/
Fedora Commons	Cornell University and the University of Virginia Library	Apache License, Version 2.0	http://www.fedora-commons.org/
Greenstone	University of Waikato	GNU General Public License	http://www.greenstone.org/
EPrints	University of Southampton	GNU General Public License	http://www.eprints.org/
Invenio	CERN Document Server Software Consortium	GNU General Public License	http://invenio-software.org/

Table 1: The five repository software systems selected for the comparative study

Using these restrictions we selected the five (5) widely used repository systems shown in Table 1. These systems are between the 10 most used open-source software participating in the Directory of Open Access Repositories – OpenDOAR (n.d.). Each of these systems has been thoroughly studied based on the core repository characteristics and supported features. We used the latest software versions which are (at August 2013): DSpace 3.2, Fedora 3.6, Greenstone 3, EPrints 3.3 and Invenio 1.1. In the following section, the characteristics needed by a current repository software are listed and described. In the third section, the five repository systems are compared based on each of the characteristics and the results are summarized in a score table. In the fourth section, one or two repository systems are proposed for the hosting of each of five different collection types.

REPOSITORY SOFTWARE CHARACTERISTICS

In our approach the essential characteristics and features that are expected from a modern repository software are analyzed for each system. The following 14 characteristics are selected based on models for repository and DL systems, like the Reference Model for an Open Archival Information System (CCSDS, 2012) and DELOS DL Reference Model (Candela et al., 2007).

1. *Object model*. The internal structure of the digital object, which according to Kahn and Wilensky (2006) is the entity that integrates metadata, digital content and relationships with other objects. Existence of unique identifiers for the digital object and every part of it is also important to ensure preservation and easy access. Possible support of digital

object typing, as denoted in (Saidis et al., 2006), that provide the manipulation and behavior of objects of the same type in a uniform manner

2. *Collections and relations support*. Collection description metadata, definition of collection hierarchies and templates that describe the format of the digital objects or the presentation of the collection. Definition of relationships between objects of the same or different types using existing standards (like RDF tuples) or custom solutions.
3. *Metadata and digital content*. Storage capabilities of the system for preserving the digital object, the metadata sets and the digital content. It is important for the repository to ensure standard as long as user defined metadata sets and multiple formats of digital content.
4. *Indexing, search and browse*. The mechanisms used for indexing and searching on the metadata. It is important for the repository to support indexing and searching not only for a restricted metadata set, but also for specified metadata fields. Support for browsing a collection based on given metadata fields and usage of controlled vocabularies are also needed.
5. *Object management*. Methods and user interfaces provided from the repository to manipulate metadata and digital content using CRUD (Create, Read, Update, Delete) actions. Support for the submission of digital objects using workflows.
6. *User interfaces*. Provided web or desktop user interfaces used as the front-end of the repository, presenting the hosted collections and the contained digital objects.
7. *Access control*. Support for users, groups and roles, as long as authentication and authorization methods. Security policies based on different granularity levels (repository, collection, digital object and content).
8. *Multiple languages support*. Multiple languages should be supported in the user interface, the metadata fields and the digital content. The character encoding is of great importance in order for the repository to be fully multilingual.
9. *Interoperability features*. Standards that the repository systems support in order to ensure interoperability with other software applications like RSS, Atom, SWORD (Simple Web-service Offering Repository Deposit) (SWORD V2 Specifications, n.d.) and others. Export of the digital objects in open standard formats is also important. In addition the use of web services assure the proper interoperability with other applications.
10. *Level of customization*. Customization of the repository 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 systems.
11. *Extended Services Support*. Extra services that are provided from the repository framework or support of plugins and add-ons from other contributors.
12. *Preservation Support.* Support for features and processes responsible for the preservation of content, including backup, replication and migration processes, preservation metadata, versioning, checksums and archiving solutions as stated in (Madali, Barve & Amin, 2012).
13. *Installations / Community Support.* The support provided by a software community is a great factor for the selection and usage of a repository software. Also a large number of installations and an active community of users and developers usually warranties the

software's evolution in the future.

14. *Collection Hosting / Cloud Support*. Many organizations provide their repository software as a service for a yearly or monthly fee. The hosting is mainly offered per collection and usually in a cloud environment.

REPOSITORY SOFTWARE COMPARISON

In this section the five repository software systems are compared based on the characteristics identified in the previous section. The level of support for each characteristic and specific considerations for each system are discussed. In Table 2, at the end of section, the five repository systems were rated based on the support for each characteristic. The minimum score for no support of a characteristic is 1 and the maximum (for full support) is 5.

Object model

DSpace: The basic entity in DSpace is *item*, which contains both metadata and digital content. Qualified Dublin Core (DC) 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, collections and communities based on CNRI Handle System (Sun, Lannom, & Boesch, 2003). 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 was initially based on Metadata Encoding and Transmission Standard (METS) (Library of Congress, n.d.). Digital object contains metadata, digital content and relationships with other objects (all treated as datastreams). 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. Fedora uses Content Model Objects for object typing. Each digital object may belong to one or more content models which define the datastreams and Service Definitions that are permitted for it.

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.

EPrints: The basic entity in EPrints is the *data object*, which is a record containing metadata. The main data object is EPrint Object representing a single record with zero or more Document Objects (digital content) attached to it. Each data object has a unique identifier and belongs to a deposit type (book, article, image, etc).

Invenio: Basic entity in Invenio is the *record* that contains metadata and may be associated with one or more documents (the digital content). Document can be stored in one or more revisions and a revision in one or more formats. Each record contains 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 only one is defined as 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 name and the description. It supports relationships between different items, using Digital Repository Interface (DRI).

Fedora: Fedora organizes objects into collections using RELS-EXT datastream. In this datastream the relationships between digital objects (like isMemberOfCollection or isPartOf) are expressed using Resource Description Framework (RDF). A default set of common relationships is defined in the Fedora relationship ontology. The relationship datastreams are indexed using the RDF-based Resource Index and a graph of all the objects and their relationships is created.

Greenstone: A collection in Greenstone defines a set of characteristics that describe its functionality. These characteristics are: indexing, searching and browsing capabilities, file formats supported, 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.

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

Invenio: Records in Invenio are organized in collections (regular and virtual) using a hierarchical structure. Collections can be customized in order to have different web interfaces, workflows and other features. Linking rules are defined in order to implement relations between documents.

Metadata and digital content

DSpace: DSpace stores by default qualified DC metadata (in PostgreSQL or Oracle database), administrative metadata and structural metadata (information about how to present an item or bitstreams). Other metadata sets and digital content are represented as bitstreams and are stored on filesystem or Storage Resource Broker (SRB). It also supports versions of items, preserving the current state of metadata, bitstreams and resource policies attached to the item.

Fedora: Metadata and digital content are both versioned and are considered datastreams of the digital object. FOXML objects and datastreams are stored using low level storage plugins including the server filesystem, Amazon's Simple Storage System (S3) or Akubra BLOB storage. While Dublin Core is the default metadata set stored in a reserved datastream, more metadata sets can be concurrently used in other datastreams in XML format. Different file formats can be stored as separate datastreams in a digital object.

Greenstone: Both documents and resources are stored on filesystem. Predefined metadata sets exist, as Dublin Core and RFC 1807, but also new metadata sets can be defined using Greenstone's Metadata Set Editor. Metadata are stored in documents using an internal XML format.

EPrints: Metadata fields in EPrints are user-defined. There are many different types of metadata fields that control how it is rendered, indexed, searched etc. The data object, containing metadata, is stored in a MySQL database and the documents (digital content) are stored on filesystem.

Invenio: MARC is the standard metadata schema used in Invenio, but other metadata sets can also be defined. Documents are stored in the filesystem. Invenio can manage multiple formats and multiple revisions for the same document.

Indexing, search and browse

DSpace: Provides indexing for the default metadata set or other defined metadata sets using Lucene or Solr search engines. They support fielded search, stemming and stop words removal. The full text of specified file formats is also extracted upon submission and indexed. Browsing is offered by default on title, author, date or subject indices and control vocabularies are supported. Searching and browsing can be limited in a collection or community.

Fedora: Default indexing is provided for the DC metadata set and digital object's system metadata. For those fields indexing and searching is managed from a relational database using constraints on a combination of fields. A generic search (gSearch) is also provided (using Lucene, Solr or Zebra search engines) that supports indexing of specified text datastreams. In addition, relationships among digital objects are indexed and are searchable using the Resource Index Search Service (RISearch) with RDF query languages (as SPARQL, iTQL and SPO). A 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 the whole document. Stemming and case sensitive searching is also available. Managing Gigabytes (MG) open-source application is used to support indexing and searching. Browsing catalogs can be defined for specific fields using hierarchical structure.

EPrints: Indexing is supported for all metadata fields, using the MySQL database indices. Full text indexing is also 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). Subject hierarchies and authority files are supported.

Invenio: Indices can be configured to support specific metadata fields and to apply stemming and remove stop words for each language. Also the full text of specified file formats is extracted and indexed. Invenio supports a search engine that combines metadata and fulltext search in a simple Google-like query language. Advanced queries can be executed using regular expressions, range of field values (e.g. dates) and selection of metadata. The results can be sorted using specific fields and ranked with respect to word similarity, number of citations or number of downloads. Also keyword taxonomies and thesauri are used. Browsing is provided per collection or type of content (e.g. books, theses, photos, etc).

Object management

DSpace: Items in DSpace are created using the web submission interface or the batch item importer, which ingests a package file (e.g. an archive file) and creates items. In both cases a workflow process may initiate depending on the collection configuration. The workflow can be configured to contain one to three steps, where different users or groups may participate to the item submission (accept/reject item or edit metadata). DSpace also provides a batch metadata editing tool. Collections and communities are created using the web user interface.

Fedora: Creation of digital objects is feasible using the Fedora Administrator (Java client or web interface) or the Directory Ingest service (using Submission Information Packages). Metadata addition or editing is provided through a text editor in Administrator. The same client is used for addition and removal of digital content. Also the Fedora Management service defines an interface in SOAP for administering the repository, including creating, modifying, and deleting digital objects or datastreams.

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

EPrints: A default web user interface is provided for the creation and editing of objects and workflows can be configured. 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).

Invenio: Objects can be submitted by an author or a librarian, through custom and fully configurable web interfaces. Workflows can be customized to create the proper steps for submission, review, conversion of documents, approval etc. Alternatively metadata and files can be ingested using customized conversion scripts, harvested from OAI-PMH compatible repositories or sent by e-mail.

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 default searching interface is provided that allows the user to search using keywords.

Fedora: The web interface of Fedora provides administration operations and 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. Service digital objects define the presentation or manipulation methods of datastreams. A DC metadata viewing page and an image manipulation applet are provided as default services.

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 can be customized based on specified XSLTs.

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. Templates are used to generate the basic layout of the pages in the repository.

Invenio: The web user interface provides browsing, searching, submitting items and using personalized features (user basket, alerts, comments, etc). It can be customized using a template system, which allows the creation of different representations based on record-based rules. Each collection can be customized for the general look and feel of its web pages.

Access control

DSpace: It supports users (e-people) and groups that hold different rights. Authentication is provided through user passwords, IP addresses, X.509 certificates, LDAP or Sibboleth protocol. Authorization system is based on associating actions with objects and the lists of e-people who can perform them. Access control rights are stored 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 (eXtensible Access Control Markup Language) policies (OASIS, 2005). Additionally it uses Fedora Security Layer (FeSL) that provides hierarchical enforcement of access control policies. Access control can be set at the collection, object or datastream level. In XACML implementation, it specifies actions at the level of management and access API, while FeSL supports in addition a set of simple CRUD actions. Authentication is provided through LDAP or user passwords.

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.

EPrints: Registered users in EPrints belong to a type (usually administrator or editor) and are able to create and edit objects. Users are authenticated using LDAP, Sibboleth protocol or simple login credentials.

Invenio: Using the Role Based Access Control (RBAC) module roles are defined, users are attached to roles and rights are granted to perform actions (such as “view restricted collections”) in collection or record level. The rights are based on user group membership or user IP address. Invenio offers user registration as long as integration with the organization's user database to authenticate users and to exploit the available user details.

Multiple languages support

All the repository systems use Unicode character encoding, so the support of different languages is provided. Every system can use multiple languages in the metadata fields and digital content. All systems, except Fedora, provide multilingual interfaces already translated in many languages. In addition, EPrints provides an XML attribute on metadata fields to define the language used for the field value.

Interoperability features

DSpace: A DSpace repository serves as a OAI-PMH Data Provider using OAICat framework. It supports crosswalk plugins that are able to translate between DSpace's object model and an external representation (e.g. MODS or METS). DSpace also supports RSS feeds, OpenURL protocol (OCLC, 2004) providing links for every item page and SWORD protocol that allows the remote deposit of items into the repository. DSpace also uses persistent URIs to access the digital content, providing a unified access mechanism to external services.

Fedora: It serves as an OAI-PMH Data Provider. Fedora is able to export digital objects as METS XML files. Supports SWORD protocol, Atom Syndication Format as a serialization of digital objects and RSS feeds. It provides two SOAP APIs: Access API for accessing digital objects and Management API for administering the repository. The Fedora REST API also exposes a subset of the two APIs as a RESTful web service.

Greenstone: Greenstone serves as an OAI-PMH Data Provider and also supports Z39.50 protocol (The Z39.50 Document, n.d.) for executing queries based on specific metadata sets. RESTful URLs and a SOAP interface are available.

EPrints: It serves as an OAI-PMH Data Provider. EPrints exports data objects in many formats among them METS, MPEG-21 Digital Item Declaration Language and BibTeX. It also supports SWORD, Atom and RSS feeds.

Invenio: It serves as an OAI-PMH Data Provider. Invenio exports records in MARCXML and BibTeX format and supports RSS feeds.

Level of customization

DSpace: The web interface is configurable using JavaServer Pages (JSP) technology or the Apache Cocoon framework (XMLUI). Indices and workflows are also customizable. DSpace supports the use of plugins (as the Packager and Crosswalk plugins).

Fedora: In Fedora every digital object can follow one or more content models that describe its format and the relationships with other objects. It is also possible to provide multiple service operations that determine the access and manipulation methods of the digital object. The operations that are available in a digital object are defined by the Service Definition Object in the associated content model. These characteristics result in a fully customizable repository. The user interface, although by default is poor, is fully customizable based on the provided REST and SOAP APIs.

Greenstone: It provides customization for the presentation of a collection based on XSLTs and agents that control specific actions of the repository. 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.

EPrints: The data objects are customized because they contain user defined metadata configured using many properties. Plugins can be written in order to export or import data objects in different text formats and templates are used for the web pages layout. A Core API in Perl is provided for developers who prefer to access basic repository functionality.

Invenio: It supports fully customizable web interfaces (configure the record view per collection), indices, search interfaces, workflows. It is implemented in a modular manner and plugins can be developed to provide extra features.

Preservation Support

DSpace: Each bitstream is associated with one Bitstream Format, which is a unique and consistent way to refer to a particular file format. A support level is defined for every bistream format, indicating the level of preservation for the specified file format. The PREMIS Schema (PREMIS Editorial Committee, 2012) is used to represent technical metadata about bitstreams. It supports the use of checksums and versioning of bitstreams. DSpace can backup and restore all of its contents as a set of Archival Information Package (AIP) files.

Fedora: Basic technical metadata are stored for each datastream ensuring content preservation. Also Fedora provides the capability of computing and storing checksums for datastreams and using that checksum to verify that the contents of that object has not been altered. It also supports the ability to version content of objects. Using the multicast journal transport, it is possible to achieve replication or read-only mirroring of Fedora servers.

Greenstone: There is no preservation support.

EPrints: The History Module records changes of objects by updating its preservation metadata. All the files and metadata comprising an object can be exported as a package (METS and MPEG-21 DIDL export plugins).

Invenio: Supports document checksums for integrity checks, multiple revisions and automatic format conversions.

Extended Services Support

Dspace: Many extensions and add-ons are available for DSpace like Dublin Core Meta Toolkit, Embargo, Joomla extensions, pluggable Storage, Semantic Search. Batch import for bibliographic formats (EndNote, BibTeX, CSV) is provided. A supervision order system exists that binds groups of users (thesis supervisors) to an item in someone's pre-submission workspace or for collaboration between researchers.

Fedora: Fedora Service Framework contains services that offer extra functionality to Fedora Repository. Such services are: Directory Ingest Service, Generic Search Service, OAI Provider Service and SWORD Service. There are also many Fedora Commons projects except Fedora Repository like Mulgara, Topaz and Fedora Middleware project. Some other services independently developed are Saxon XSLT Processor Service, the FOP Service (provides a PDF transformation service) and the Image Manipulation Service.

Greenstone: Uses plugins for processing different file formats as MS Word, PDF, Postscript, HTML, BibTeX and extracts appropriate metadata. Plugins are used to ingest externally-prepared metadata in different forms as XML, MARC, CDS/ISIS, ProCite, BibTeX, OAI, DSpace and METS.

EPrints: Using Export plugins the repository document objects can be exported in many different formats (METS, MODS, MPEG-21 DIDL, BibTeX, EndNote, etc). Other plugins that are implemented are Import plugins, Screen plugins and Convert plugins.

Invenio: It supports personalization services, like user-defined document baskets and automated email notification alerts, as long as collaboration services, like basket-sharing within user groups. Citation statistics are offered for papers hosted in a repository. BibFormat module provides output formats like Dublin Core, EndNote, NLM, RSS and Google Scholar. WebJournal module allows to display the records of an installation in the form of an online journal.

Installations / Community Support

DSpace: It has over a thousand known installations and the largest number of repositories participating in OpenDOAR. Community of users and developers is very active. The developers involved are categorized as committers and code contributors, while an advisory team is responsible for new releases and new features of DSpace. Support for DSpace users is provided using the mailing lists, web seminars, training materials, presentations and user manuals. Users can also submit bugs and issues for the most current release of the software. The community works with the help and guidance of DuraSpace organization. DuraSpace Registered Service Providers offer commercial support for DSpace installations.

Fedora: It has an active community of developers involved in the development of Fedora Commons projects (as commiters or code contributors), as long as other software designed to work with Fedora Repository. Support is provided using the user mailing list, community articles (how-to guides, FAQ, white papers, etc) and user manuals. Fedora community is also coordinated by DuraSpace. DuraSpace Registered Service Providers offer commercial support for Fedora installations.

Greenstone: There are over 80 known installations on libraries, universities and NGOs, mostly in developing countries. It has a community of developers that contribute their code for Greenstone software. Community support is provided using mailing lists, documentation wiki, FAQ, user manuals and self-study courses. Commercial support is provided by some collaborative companies (repository customization, technical support and maintenance).

EPrints: Over 350 installations are registered in OpenDOAR. Developers contribute in the development of the core software (mostly from dev team), plugins, themes and translations. Community support provided using training material, tutorials, mailing lists, wiki, manuals and how-to guides. EPrints Services team offers a commercial-like technical support, repository customization and training.

Invenio: More than 30 known installations exist, mostly on research organizations and universities. Development is realized by a specific team of developers. Community support is provided through community mailing lists and chatroom, as long as by the installation and administration guides. Commercial support is also offered using a special collaboration contract with the CERN development group.

Collection Hosting / Cloud Support

DSpace: DSpaceDirect is a hosted service of DSpace repository software in the cloud, offered by DuraSpace.

Fedora: Shared or dedicated hosting of Fedora repository is provided from third party organizations.

Greenstone: Shared or dedicated hosting of collections in a Greenstone repository is provided from commercial organizations.

EPrints: Repository hosting and maintaining is offered by the EPrints Services team. There is also a downloadable Amazon Machine Image (AMI) including a Debian linux system with EPrints pre-installed for Amazon Elastic Compute Cloud (EC2).

Invenio: No collection hosting is provided for Invenio software in a subscription basis, but Invenio-based OpenAIRE Orphan Record Repository supports the hosting of research articles if there is no access to an OpenAIRE compliant repository.

Characteristics	*DSpace*	*Fedora*	*Greenstone*	*EPrints*	*Invenio*
Object model	4	5	3	4	3
Collection and relations support	4	4	5	2	4
Metadata and digital content	4	5	4	4	4
Indexing, search and browse	4	4	4	4	5
Object management	4	2	2	4	5
User interfaces	4	2	3	4	4
Access control	5	4	2	3	4
Multiple languages support	4	3	4	5	4
Interoperability features	4	5	3	4	3
Level of customization	3	5	3	4	3
Preservation Support	4	4	1	3	3
Extended Services Support	4	5	3	3	5
Installations / Community Support	5	4	4	5	3
Collection Hosting / Cloud Support	5	4	4	4	3

Table 2: Level of support for the characteristics of each repository

DIGITAL COLLECTIONS CASE STUDIES

In the following paragraphs, five different collection types are described and one or two repository software systems are proposed in each case. The software is proposed based on the special features specified by each collection as long as the flexibility that the system provides in order to implement some of the features. We selected five collection paradigms to represent different needs and features, regarding metadata, digital content format, relationships, administration and preservation issues. The collections are: a scientific data collection, a digitized content collection, a rare books collection, an Electronic Theses and Dissertations (ETDs) collection and a new media art collection. For each case we state a brief description, a list of the content types supported, a specification of the special features required and we propose the repository software for the collections hosting.

Scientific data collection

Case description: Scientific data extracted from research experiments, observations or surveys usually are critical and valuable data, important for researchers worldwide. For many years scientific data used to be stored in local databases or custom applications developed by the research organizations, because scientific repositories and DLs was not always high on the priority list of science and technology researchers (Wallis et al., 2010). On the other hand, Digital Agenda for Europe (2010) states that publicly funded research should be widely disseminated through Open Access publication of scientific data and papers. Scientific repositories are needed for managing and sharing datasets, publications, reports and other types of content for public or internal use. Researchers should have the ability to submit their datasets or publications and select the access policies.

Content types: Datasets mostly in text files or spreadsheets, documents (usually Word documents and PDFs), presentations. Sometimes video and audio files from observations are available.

Special features: Submission by researchers and curation by librarians or specialized staff. User registration support and access policies. Linking between objects (datasets and publications). Exporting datasets in common formats.

Proposed solution: For this case it seems that the most appropriate software systems are Invenio and DSpace. They support workflows where a registered user may proceed with the submission and other user groups may review and edit the submitted object. Furthermore they support linking between objects of different collections (e.g. publication object with dataset object). In addition, they provide collaboration features for user groups and Invenio supports citation metrics for articles. Paradigms of such repositories are Zenodo Repository (n.d.) and CERN Document Server (n.d.) which are implemented using Invenio software and Dryad Repository (n.d.) developed using DSpace.

Digitized content collection

Case description: An organization is planning to digitize collections from libraries, archives and museums and host them in a single repository. The organization has human resources and the amount of time in order to customize the DR 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.

Content types: Images, videos and 3D objects of digitized items (books, paintings, objects, sculptures, etc).

Special features: Different metadata sets and digital content formats. Relationships between objects. Submission by librarians, archivists or curators. Preservation support. Detailed access policies depending on content.

Proposed solution: In that case the most suitable repository system is Fedora, since it provides a very customizable modular architecture. It supports multiple collections having different content models, various content formats that may be associated with proper service objects for their presentation and manipulation. Preservation features as technical metadata, versioning, checksums and content replication are supported. Access policies can be defined using XACML or FeSL in collection, object or datastream level. Submission of objects is feasible using the Fedora Administrator but using REST or SOAP APIs collection specific web interfaces can be developed. An example of a repository that hosts multiple digitized collections using Fedora is Pergamos Digital Library (Pyrounakis et al., 2006).

Rare books collection

Case description: A library plans to electronically publish rare books in an easy to use customizable repository system. The books are digitized as high quality images and their structure must be retained based on the book's table of contents. The full text of each book is extracted and should be searchable. Basic metadata will be stored for each book as title, author and publication year. The library does not possess enough human resources for the installation and customization of the repository, so it needs an “out of the box” solution.

Content types: Digitized images of the book pages, extracted text and PDF files.

Special features: Submission of content and metadata by the librarians. Hierarchical structure of books. Full text indexing of book content. Easy installation and maintenance.

Proposed solution: In that case the most appropriate repository system is Greenstone, since by default it represents books in a hierarchical manner, using table of contents. The full text of the book is searchable in paragraph, chapter or document level, using the provided search engine. Greenstone requires few human resources for its installation and maintenance, because it is designed and developed considering its distribution to organizations in developing countries.

Electronic Theses and Dissertations collection

Case description: A university needs a digital repository for ETDs as long as for publications produced by students and staff. Documents are submitted by authors and staff, using basic metadata and predefined subjects. The hierarchy of the organization should be represented in the repository. The collections will be part of a federated repository for ETDs using OAI-PMH.

Content types: Mostly documents in Word or PDF format and archive files containing additional data.

Special features: Use of authority files. Users authenticated using LDAP or other centralized authentication protocol. Simple web interfaces for the submission of documents. OAI-PMH Data Provider support.

Proposed solution: In that case, the most appropriate systems are EPrints and DSpace. They both use authority files to implement subject headings and support the organization's hierarchical structure (DSpace by default represents communities and EPrints hierarchical authority files). They provide web interfaces for the submission of metadata and digital content by registered users. The users can be authenticated by the organizations' authentication mechanism using LDAP or Sibboleth protocol. Both repositories support OAI-PMH as Data Providers, so they can contribute to the federated repository.

New media art collection

Case description: An artistic institute plans to host collections of multimedia objects (photos, videos, animation, music) in a digital repository. The repository will host digital objects mainly from new media artists (computer animators, video artists, photographers, etc). The submission of objects should be feasible in an easy manner using a web interface or a Dropbox-like desktop client in order to facilitate artists to upload their work. Users should have the ability to register, built their own collections and submit their works of art using simple metadata (title, description, date, media category). The web interface should use the proper media players to display properly the different media forms (video and audio player, image slideshow, etc).

Content types: Image, video and audio files.

Special features: Web interface and desktop client for submission. User registration. Advanced security policies. Creation of collections and upload of digital content by the registered users. Media players for the different content types.

Proposed solution: In order to develop desktop client applications for the submission to the repository, a proper API should be provided. As stated in (Lewis et al., 2012) SWORD can be used in many cases for developing alternative deposit applications. Especially SWORD v2 that allows CRUD actions on the repository is an ideal solution. DSpace and Fedora support this version of SWORD and also support the usage of collections. The web interface for the media presentation may also use SWORD protocol for reading digital content and metadata. These two repository software systems support security policies to allow each user to edit his/her collections of objects.

CONCLUSION

The open-source DR software developed the last decade has reached a mature level. The software systems proposed in this paper are the result of many years of development and usage experience. They all cover the core functionality of a contemporary repository software, are consistent and user friendly. They provide efficient support from the developers community and improved installation processes. Some of them are provided as “out of the box” solutions, while others need specialized IT personnel to get involved for its installation and customization.

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 repository. In each case the specialized staff should be involved in the customization of the software in order to manage the best results. As it was expected not all software is suitable for each case. On the other side there is no single software solution that matches all criteria. Each system has its advantages and drawbacks, as stated in the comparison. This comparative study cannot present a unique solution but can be used as a guideline for organizations planning to host digital collections or migrate their collections to a new repository environment.

REFERENCES

Candela, L. (2007). *The DELOS Digital Library Reference Model – Foundations for Digital Libraries Version 0.98*. DELOS. Retrieved from http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf

CERN Document Server. (n.d.). Retrieved August 10, 2013, from http://cds.cern.ch/

Consultative Committee for Space Data Systems (CCSDS). (2012). *Reference Model for an Open Archival Information System (OAIS) Issue 2*. Retrieved from http://public.ccsds.org/publications/archive/650x0m2.pdf

Digital Agenda for Europe. (2010). Retrieved from http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0245:REV1:EN:HTML

Directory of Open Access Repositories - OpenDOAR. (n.d.). Retrieved August 10, 2013, from http://opendoar.org/

Dryad Repository. (n.d.). Retrieved August 10, 2013, from http://datadryad.org/

Goh, D. H.-L., Chua, A., Khoo, D. A., Khoo, E. B.-H., Mak, E. B.-T., & Ng, M. W.-M. (2006). *A checklist for evaluating open source digital library software.* Online Information Review, 30(4), 360–379.

Kahn, R., & Wilensky, R. (2006). *A framework for distributed digital object services. International Journal on Digital Libraries*, 6(2), 115–123.

Lagoze, C., & Van de Sompel, H. (2001). *The open archives initiative: building a low-barrier interoperability framework*. In Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries (pp. 54–62). New York, NY, USA.

Lewis, S., de Castro, P., & Jones, R. (2012). *SWORD: Facilitating Deposit Scenarios*. D-Lib Magazine, 18(1/2).

Library of Congress. (n.d.). *METS: An Overview & Tutorial*. Retrieved from http://www.loc.gov/standards/mets/METSOverview.v2.html

Madalli, D. P., Barve, S., & Amin, S. (2012). *Digital Preservation in Open-Source Digital Library Software*. The Journal of Academic Librarianship, 38(3), 161–164.

Masrek, M. N., & Hakimjavadi, H. (2012). *Evaluation of Three Open Source Software in Terms of Managing Repositories of Electronic Theses and Dissertations: A Comparison Study*. Journal of Basic and Applied Scientific Research, 2(11), 10843–10852.

OASIS. (2005). *eXtensible Access Control Markup Language (XACML) Version 2.0*. Retrieved from http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

OCLC. (2004). *The OpenURL Framework for Context-Sensitive Services* (NISO Standard). Retrieved from http://www.niso.org/apps/group_public/download.php/6640/The%20OpenURL%20Framework%20for%20Context-Sensitive%20Services.pdf

Open Society Institute. (2004). *A Guide to Institutional Repository Software 3rd Edition*. Retrieved from http://www.budapestopenaccessinitiative.org/resources/guide-to-institutional-repository-software

PREMIS Editorial Committee. (2012). *PREMIS Data Dictionary for Preservation Metadata Version 2.2*. Retrieved from http://www.loc.gov/standards/premis/v2/premis-2-2.pdf

Pyrounakis, G., Saidis, K., Nikolaidou, M., & Karakoidas, V. (2006). I*ntroducing Pergamos: A Fedora-Based DL System Utilizing Digital Object Prototypes*. In Research and Advanced Technology for Digital Libraries (pp. 500–503). Springer Berlin Heidelberg.

Saidis, K., Pyrounakis, G., Nikolaidou, M., & Delis, A. (2006). *Digital Object Prototypes: An Effective Realization of Digital Object Types*. In Research and Advanced Technology for Digital Libraries (pp. 123–134). Springer Berlin Heidelberg.

Sun, S., Lannom, L., & Boesch, B. (2003). *Handle System Overview (RFC 3650)*. Corporation for National Research Initiatives. Retrieved from http://www.ietf.org/rfc/rfc3650.txt

SWORD V2 Specifications. (n.d.). Retrieved August 10, 2013, from http://swordapp.org/sword-v2/sword-v2-specifications/

The Z39.50 Document. (n.d.). Retrieved August 10, 2013, from http://www.loc.gov/z3950/agency/document.html

Tramboo, S., Humma, H., M Shafi, S., & Gul, S. (2012). *A study on the Open Source Digital Library Software: Special Reference to DSpace, EPrints and Greenstone*. International Journal of Computer Applications, 59(16), 1–9.

Wallis, J. C., Mayernik, M. S., Borgman, C. L., & Pepe, A. (2010). *Digital libraries for scientific data discovery and reuse: from vision to practical reality*. In Proceedings of the 10th annual joint conference on Digital libraries (pp. 333–340). New York, NY, USA.

Zenodo Repository. (n.d.). Retrieved August 10, 2013, from https://zenodo.org/