Repository Applications

This section looks at repository software applications available for use by cultural heritage institutions for preservation functions. There are, of course, some operating repositories, such as the University of California's Digital Preservation Repository and the Stanford Digital Repository, which use sophisticated, locally developed systems that are not available for general use. These are not covered in this section.

Most available systems, including DSpace, EPrints, and Fedora, are institutional repository (IR) applications designed to collect and disseminate the intellectual output of a university or other academic institution. DAITSS and LOCKSS differ from the IR applications in that preservation is their primary function. DAITSS is a "dark archive," a preservation repository with no end-user interface, built strictly along the OAIS model. LOCKSS is an automated mechanism for harvesting and ensuring the integrity of Web-accessible content, primarily e-journals. aDORe is a solution to the storage component of a preservation repository.

Institutional Repositories

Institutional repositories trace their roots to disciplinary e-print servers established to facilitate early and open access to research literature by allowing authors to self-archive digital preprints or postprints of their own journal articles. Topical aggregations were necessary because each repository was a silo with its own search-and-retrieval system and there was no easy way for researchers to search multiple repositories at once. This situation changed with the 1999 release of the Santa Fe Convention, which subsequently became the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Repository applications that supported OAI could export their metadata for harvesting into other aggregations, essentially uncoupling the repository itself from the discovery function. This allowed universities to establish institution-based repositories without disrupting disciplinary research.

The movement for institutional repositories has been carried along by a whole new set of goals. For some, they are a way for the university to promote itself by collecting and disseminating the works of its faculty. For others, they are a means of undermining traditional scholarly publishing and a weapon in the battle against skyrocketing journal subscription prices. To many, *institutional repository* is synonymous with *preservation repository.* An often-quoted article by Clifford Lynch calls the IR "most essentially an organizational commitment to the stewardship of these digital materials, including long-term preservation where appropriate, as well as organization and access or distribution."[1] Unfortunately, in the United States, institutional repositories are underused and struggling to attract content. A 2007 study counted roughly 100 IRs in U.S. universities and colleges, but also found the median growth rate was only one item a day.[2] Growth occurs when deposit can be mandated—for example, for electronic dissertations—or done by administrative staff, as for departmental publications. Faculty use, however, is nearly always voluntary and nearly always low.

Most institutional repository systems support the same set of core functions. They provide a mechanism for submitters to register, and for registered submitters to log on. They provide a mechanism for uploading digital materials and forms for entering descriptive metadata, whether by the (presumably bibliographically unskilled) author or some other party. They allow submissions to be reviewed and edited before approval for ingest. They ingest, store, and provide some level of content management for the digital materials. They provide access to the materials using local search-and-retrieval features and support some methods of access control. They have the ability to expose OAI-PMH–compliant metadata for harvesting into other aggregations.

Institutional repositories may not be complete preservation solutions, but they do make preservation possible by allowing responsible curatorial organizations to capture content they might otherwise not have, and to secure appropriate rights from the authors. Viewed as preservation repositories, IRs have ingest and metadata capabilities that support the Availability and Identity goals of the core preservation activities. Because the content is managed centrally by skilled IT staff, the Fixity and Viability of the stored files should also be ensured. Where IR applications differ is in the extent of their support for Authenticity and Renderability.

DSpace, Fedora, and EPrints are the three dominant applications used for institutional repositories, although each can be used as a platform for other services as well. Each incorporates, or has plans to incorporate, different preservation functionality as noted below. DSpace, Fedora, and EPrints are also the focus of the annual Open Repositories conference, which features open user-group meetings for the three applications, followed by sessions on topics of common interest, often including preservation. Papers from Open Repositories 2007 are available online. The 2008 meeting will be held in Southampton, England, in April.

Open Repositories 2007
http://openrepositories.org/2007

Open Repositories 2008
http://or08.ecs.soton.ac.uk

Readings

- Susan Gibbons, "Establishing an Institutional Repository," *Library Technology Reports,* July/Aug. 2004, https://publications.techsource.ala.org/products/archive.pl?article=2538. The description of particular software applications is a bit dated, but the background information remains solid.
- "Technical Evaluation of Selected Open Source Repository Solutions," version 1.3, 2006. https://eduforge.org/docman/view.php/131/1062/Repository%20Evaluation%20Document.pdf. Compares DSpace, EPrints and Fedora for a project in New Zealand.
- *Open*DOAR: The Directory of Open Access Repositories Web site, www.opendoar.org. An authoritative, well-maintained international directory of academic institutional repositories.
- Karen Markey et al., *Census of Institutional Repositories in the United States: MIRACLE Project Research Findings* (CLIR pub 140, Feb. 2007), www.clir.org/pubs/abstract/pub140abst.html.

DSpace

DSpace is the most widely used institutional repository application in the United States. It was developed jointly by the MIT libraries and Hewlett-Packard and first released in 2002. In 2007 the DSpace Foundation was established as a nonprofit organization to support DSpace users and coordinate future development.

DSpace runs on any flavor of Unix. It is written in Java and JSP, and can use either PostgreSQL or Oracle. Source code is available under the BSD license. Development is distributed along the Apache model, where authorized developers from the user community can submit code to be integrated into the main code base by a small team of committers.

DSpace was developed from scratch to be usable in university environments, where there are many autonomous departments with their own governance and cultures. Submission functions are designed around "communities," or administrative units, which can have their own metadata templates and ingest rules, and can set up their own collections. DSpace can ingest any digital format, and supports qualified Dublin Core metadata. It can export content and metadata according to a local XML schema.

It is hard to discuss features of DSpace because there is an active group of open-source developers working on a wide variety of programming projects. Some of these projects have been very significant. For example, the difficulty of configuring the user interface was a major problem until the Manakin project at Texas A & M developed an XML-based user interface, which has been very favorably received. A complete list of projects can be found on the DSpace wiki. In addition, a major architectural revision, DSpace 2.0, has been in the planning stages for some time.

The main preservation feature of DSpace is the association of formats with one of three preservation levels: supported, known, and unsupported. Supported formats are those for which renderability will be ensured by format migration or emulation (although this is a statement of intent only; DSpace does not include migration or emulation routines). Known formats are "those that we can't promise to preserve . . . but which are so popular that third party migration tools will likely emerge to help with format migration."[3] Unsupported formats will be subject to bit-level preservation only.

Community projects to extend the preservation functionality of DSpace are encouraged. The Digital Preservation Tools and Strategies Project at Cambridge University, which ended in 2006, investigated strategies for improving the digital preservation functionality of DSpace and did some work to incorporate JHOVE for format identification and validation. Development by the University of California at San Diego allows DSpace to use the San Diego Supercomputer Center's Storage Resource Broker for bitstream storage as an alternative or supplement to the DSpace implementation's own file system. Other projects are looking at DSpace as a component in a wider preservation system.

DSpace Web site
www.dspace.org

DSpace wiki
http://wiki.dspace.org/index.php/Main_Page

Reading

- Joseph G. Pawletko and Ekaterina Pechekhonova, "A DSpace-Based Preservation Repository Design," presentation to the DLF Fall Forum 2006, www.diglib.org/forums/fall2006/presentations/pawletko-2006-11.pdf.

Fedora

Fedora (not to be confused with the Linux-based operating system of the same name) is an open-source repository application originally developed by Cornell University and the University of Virginia Library. Like DSpace, it is now managed by a nonprofit organization, the Fedora Commons.

Fedora is written in Java and requires a relational database (McKoi SQL Database, MySQL, PostgreSQL, or Oracle 9). Source code is available under the Educational Community License (ECL). To this point, development of the core repository service has been controlled by staff developers, while Fedora users have built supplementary applications using Fedora APIs. The new Fedora Commons has not yet announced its plans for partnership and participation opportunities.

Fedora implements a digital object repository architecture built around a flexible and extensible object model that manages data, metadata, and relationships. The repository runs as a service within a web server and exposes its functionality as web services. The core repository service includes ingest, validation, storage, access, dissemination, and management tools. As such, it is not actually an institutional repository system, but provides a platform on which more complete applications such as IRs can be built. Many of these applications are also available as open-source distributions.

Fez, a front-end management system developed by the University of Queensland, provides IR functionality for Fedora, including submission and other workflows, metadata with controlled vocabularies, search, browse, and security features. "Fez + Fedora" and DSpace are the two IR systems supported by the Australian Partnership for Sustainable Repositories (APSR). According to the February 2007 issue of *Sustaining Repositories,* the APSR newsletter, use of Fez + Fedora is spreading internationally:

> Emory University Libraries are building a Fez repository for electronic theses. Indiana University Libraries are also testing Fez+Fedora to see whether to replace their existing DSpace installation. The Colorado Alliance of Research Libraries is using Fez+Fedora for their Alliance Digital Repository. Also in the US, the National Science Digital Library is using Fez+Fedora for their Materials Science Digital Library.[4]

Another increasingly popular Fedora-based IR is the commercial product VITAL, developed and supported by the library systems vendor VTLS.

Part of Fedora's appeal is its ability to provide a common repository layer that can underlie many different applications. It would be possible, for example, to have a digital encyclopedia, "digital library" collections, an IR, and an electronic journal publishing system all sharing content and using Fedora as their backend store.

At the time of this writing, actual preservation support in Fedora is fairly minimal, but the Fedora Commons has an active interest in developing this area. There is a Preservation Services Working Group, which gave this report at the Open Repositories Conference 2007:

> A major objective of the Fedora Preservation Services Working Group (WG) is to define the requirements and architecture for preservation services that can be integrated into Fedora. We believe our work will provide capabilities for Fedora users to build trusted repositories. To accomplish our objectives, the Working Group is specifying services and technologies that can be readily integrated into the Fedora Framework. In the specification process, the WG is focused on the underlying capabilities to support digital object persistence, life cycle management, multidisciplinary collections, and management of the repository environment (e.g. storage, memory, operating system, etc). Capabilities and features

currently under consideration include checksum creation and validation, event management and messaging, and a repository history service.[5]

Preservation work was in hiatus for most of 2007 but is expected to start up again now that the Fedora Commons is established.

Fedora Commons Web site
www.fedora-commons.org

Fedora wiki
www.fedora.info/wiki/index.php/Main_Page

Reading

- "Fedora and the Preservation of University Records Project: 4.1 Analysis of Fedora's Ability to Support Preservation Activities," Sept. 2006, http://repository01.lib.tufts.edu:8080/fedora/get/tufts:UA069.004.001.00011/bdef:TuftsPDF/getPDF.

EPrints

EPrints, also called GNU EPrints, was developed at the University of Southampton in the United Kingdom and released in 2000. It was the first open-source institutional repository software application and is still, worldwide, the most widely used. The latest version, EPrints 3, was released in 2007.

EPrints runs on any flavor of Unix, although Red Hat is recommended, and it requires Apache 2.0, mod_perl, perl, and mySQL. Although the source code is distributed under a GPL license, development is controlled by staff at the University of Southampton.

Because EPrints was initially developed as an extension of a disciplinary repository system, it is primarily geared to e-prints of journal literature, although it can ingest digital objects in any format. It supports input and edit of simple Dublin Core metadata. It allows customization of document types, document formats, metadata fields, subject categories, workflow, and views. It supports metadata-based and full-text search. There is batch import capability as well as item-by-item deposit. Both data and metadata can be exported in a variety of formats.

Acknowledged strengths of EPrints include the ease of installation and maintenance and the well-structured and -documented code. The large user base is also cited as a factor in ensuring the continued maintenance of the software. A disadvantage at the time of this writing is that version 3 is not as stable or complete as version 2 and has introduced a number of bugs.

Preservation features of EPrints are new to version 3 and documented on the Eprints technical wiki:

- Export of complete structured complex objects is available using either METS or the MPEG-21 Digital Item Declaration Language (DIDL) as an XML container format.
- A "history" function updates and stores a history record documenting all actions performed on objects and all changes made to them.
- A rights declaration has been added to the user interface. This will allow permission for preservation actions to be recorded.

Despite these enhancements, EPrints developers have taken a different approach to digital preservation from that of DSpace or Fedora. Rather than attempting to develop EPrints into an OAIS-conformant preservation repository, the plan is make the EPrints repository one functional unit in a distributed network of preservation services. This model will be further developed by the JISC-funded Preserv 2 project.

EPrints Web site
www.eprints.org

EPrints wiki article on preservation support
http://wiki.eprints.org/w/Preservation_Support

Preserve 2 Web site
http://preserv.eprints.org

Reading

- Steve Hitchcock et al., "Digital Preservation Service Provider Models for Institutional Repositories: Towards Distributed Services," *D-Lib Magazine* 13 no. 5/6 (May/June 2007), www.dlib.org/dlib/may07/hitchcock/05hitchcock.html.

DAITSS

DAITSS was developed by the Florida Center for Library Automation (FCLA) as the software application underlying the Florida Digital Archive, a preservation repository for the use of the public universities of the state of Florida. It is available for use under the open-source GPL license, but there is no active community of developers beyond the FCLA staff. DAITSS is meant to run under Linux or Unix, is written in Java and Ruby, and uses a MySQL database.

DAITSS is designed to conform closely to the OAIS model. It will accept SIPs and transform them into AIPs for storage. As part of the ingest process, DAITSS identifies the format of each incoming file, validates the file, extracts technical metadata, and creates normalized or migrated versions if necessary. The original files as submitted are always preserved, along with the "last, best" version created by DAITSS, and both the originals and "last, best" files are disseminated in a DIP on request.

Since all format transformations are done at the time of ingest, it is possible that content archived through the system may not be stored in current renderable formats at any given point. To get around that problem, when a dissemination is requested, DAITSS exports the AIP and transforms it into an SIP for re-ingesting. As the files are re-ingested, they are identified, validated, and normalized or migrated just like any other SIP contents. The newly updated AIP is then disseminated to fulfill the request. As such, the system can be said to perform "just in time" format transformation. Mass migration can be accomplished by using the reporting system to identify all AIPs containing files requiring migration and requesting their dissemination.

DAITSS can accept content in any format, but can ensure renderability only for supported formats, which are listed on the project Web site. A DAITSS repository is most suitable for text, image, audio, and video formats, which benefit from migration, as opposed to video games and other interactive multimedia more amenable to emulation. DAITSS records PREMIS-conformant metadata and maintains digital provenance in the form of PREMIS event records. There is an administrative interface for repository staff, but no access interface for end users. The system is designed to be used as a stand-alone preservation repository or as a backend to institutional repository systems, digital library systems, e-journal publishing systems, and the like.

An advantage of DAITSS is that it was designed from the start to implement format transformation as a preservation strategy and so provides a complete infrastructure for supporting migration. A weakness is that the system is difficult to install, configure, and manage, and is very difficult to contribute code to. Development on a new generation of the software, DAITSS 2.0, began in 2007 and will hopefully remedy some of these problems.

DAITSS Web site
www.fcla.edu/digitalArchive

DAITSS wiki
http://daitss.fcla.edu

Reading

- Priscilla Caplan, "The Florida Digital Archive and DAITSS: A Working Preservation Repository Based on Format Migration," www.fcla.edu/digitalArchive/pdfs/IJDL_article.pdf.[6]

LOCKSS

LOCKSS (Lots Of Copies Keeps Stuff Safe) was developed at Stanford University as a way for libraries to capture and store their own local versions of digital content, according to the theory that the more institutions hold something, the less likely that something is to disappear. Originally designed for electronic journals, it can store any Web-accessible content. The freely available software was released in beta version in 2000 and in production in 2004.

LOCKSS is designed to run on stand-alone low-end microcomputer hardware. The installation/boot disk contains its own operating system and everything necessary to turn the machine into a "LOCKSS Box." Installation and operation are very simple.

Before a publication can be harvested by a LOCKSS Box, the publisher has to give permission in the form of a "manifest" page on the Web site, and a LOCKSS plug-in has to exist for the publishing platform. The library running the LOCKSS Box configures it with a list of titles the library wants to store. The publisher's license generally allows the library's users to access the content from the LOCKSS Box in the event the publisher's content is unavailable, for example, when the publisher's server is down, or if the library discontinues its subscription to a title.

LOCKSS ensures the fixity of the contents in an innovative way. Multiple LOCKSS Boxes with overlapping content periodically participate in "polls," where the message digest (checksum) of the content held in common is compared. If the content of one box is different from the others, it is assumed to be damaged and automatically replaced with a good copy. Details of the "majority voting, fault tolerant" system are described by Rosenthal and Reich.[7]

Although LOCKSS primarily guarantees Availability and Fixity, developers have begun to address Renderability by doing a proof of concept that forward format migration can be done at the time that stored content is delivered to the Web browser.

The LOCKSS Alliance is a library membership organization with annual fees ranging from just over $1000 to almost $11,000 depending on the size and type of institution. Alliance members get some incentives such as access to "premium" content, but participation is mostly a way of supporting the LOCKSS initiative.

Most of the publishers participating in LOCKSS are open-access publishers or university presses. In contrast, CLOCKSS (Controlled LOCKSS) is a smaller and more formal partnership involving eleven scholarly publishers (many of which do not participate in public LOCKSS), six libraries, and OCLC. By agreement with the publishers, the CLOCKSS libraries host LOCKSS Boxes comprehensively caching these publishers' journal output, and will make it available to everyone without restriction when "no publisher has current responsibility for, nor is providing electronic access to" the content.

Other projects are finding LOCKSS useful as a core or auxiliary technology. The MetaArchive of the American South and the Alabama Digital Preservation Network are using LOCKSS as the basis of a distributed preservation network infrastructure aimed at historical materials and locally created digital content, respectively. Other projects are using LOCKSS as a tool for replication and verification of file integrity as part of a larger preservation repository application.

LOCKSS Web site
www.lockss.org/lockss/Home

CLOCKSS Web site
www.clockss.org/clockss/Home

Reading

- David H. S. Rosenthal, et al., "Transparent Format Migration of Preserved Web Content," *D-Lib Magazine* 11 no. 1 (Jan. 2005), www.dlib.org/dlib/january05/rosenthal/01rosenthal.html.

aDORe

aDORe was developed at Los Alamos National Laboratory (LANL) by the Digital Library Research & Prototyping Team. Development of the software was partly funded by an NDIIP grant from the Library of Congress. It was released in 2006 for public use under the GNU Lesser General Public License (LGPL). Development is controlled by LANL. The system is written in Java and has been tested on Linux, Windows, and Mac OS platforms.

aDORe is not a complete preservation solution but is designed to address the problem of storage for digital repositories. Writing digital objects as individual files in filesystems is inefficient in terms of I/O (input/output) for storage, access, backup, and recovery. Concatenating objects together in "tar" or "zip" files is more efficient for I/O but impedes the use of standard tools for indexing and manipulating the contents. The aDORe approach is to separate the storage of data files (bitstreams) and their metadata and to package each in formats that can be easily accessed. For a given set of digital objects, the XML-based metadata files (for example, MPEG-21 DIDL or METS documents) are concatenated together in larger files called XMLTape files. The content data files themselves are combined into ARC files, a format developed by the Internet Archive (see Chapter 7). The XMLTape files and ARC files reference each other by means of identifiers.

An advantage of aDORe is that it integrates widely used open-source tools and standards. The metadata in the XMLTape can be indexed and accessed through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). A datastream in an ARC file located through the OAI-PMH repository can be accessed via its OpenURL. The architecture allows for multiple aDORe archives to be easily federated.

Another benefit is that the standards-based approach should significantly lower the learning curve for implementers. The system is documented and includes a tutorial. A plug-in architecture will allow developers to build their own customized pre-processors for the content of XMLtapes and ARC files.

aDORe Archive Web site
http://african.lanl.gov/aDORe/projects/adoreArchive

Reading

- Xiaoming Liu et al., "File-Based Storage of Digital Objects and Constituent Datastreams: XMLtapes and Internet Archive ARC files," June 2005, http://arxiv.org/pdf/cs.DL/0503016.

Notes

1. Clifford Lynch, "Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age," *ARL Bimonthly Report* 226 (Feb. 2003), www.arl.org/resources/pubs/br/br226/br226ir.shtml (accessed Nov. 17, 2007).
2. Cat S. McDowell, "Evaluating Institutional Repository Deployment in American Academe Since Early 2005 Repositories by the Numbers, Part 2," *D-Lib Magazine* 13 no. 9/10 (Sept./Oct. 2007), www.dlib.org/dlib/september07/mcdowell/09mcdowell.html (accessed Nov. 17, 2007).
3. DSpace FAQ, www.dspace.org/index.php?option=com_content&task=blogcategory&id=40&Itemid=88 (accessed Nov. 17, 2007).

4. *Sustaining Repositories: Newsletter of the APSR Partnership,* Feb. 2007, www.apsr.edu.au/news/newsletter 2007feb/newsletter.pdf (accessed Nov. 17, 2007).
5. “Accepted Presentations for the Fedora User Group Sessions,” Open Repositories Conference 2007, http://openrepositories.org/2007/program/fedora (accessed Nov. 17, 2007).
6. First published as an article in the *International Journal on Digital Libraries,* 20 March 2007, http://dx.doi.org/10.1007/s00799-007-0009-6. The original publication is available at www.springerlink.com.
7. David S. H. Rosenthal and Vicki Reich, “Permanent Web Publishing,” presentation, FREENIX Program, USENIX 2000, http://lockss.stanford.edu/freenix2000/freenix2000.html (accessed Nov. 28, 2007).