Open Society Institute

A Guide to Institutional Repository Software

3rd Edition

August 2004

Acknowledgments

The Open Society Institute and the author wish to thank the following representatives of the systems discussed in the following pages for their time, diligence, and patience in reviewing and commenting on the information presented here: Rida Benjelloun and Guy Teasdale of the University of Laval (Archimede); Erik Groeneveld of Seek You Too B.V. (I-Tor); Christopher Gutteridge of the University of Southampton (Eprints.org); Henk Harmsen and Laurents Sesink of the Netherlands Institute for Scientific Information Services (i-Tor); Jean-Yves Le Meur of CERN (CDSware); Frank Lützenkirchen of the University of Essen (MyCoRe); Thomas Place, Wilko Haast, and Fred Vos of Tilburg University (ARNO); Frank Scholze and Annette Maile of Stuttgart University Library (OPUS); MacKenzie Smith and Richard Rogers of MIT (DSpace); and Chris Wilper of Cornell University (Fedora).

Additionally, Henk Ellerman (Erasmus Electronic Publishing Initiative), Martin Feijen of Innervation (consultant to DARE), Susan Gibbons (University of Rochester), Steve Hitchcock (University of Southampton), Peter Linde (Blekinge Institute of Technology), William Nixon (University of Glasgow), Andrew Treloar (Monash University), and Lilian van der Vaart (DARE) have generously provided valuable feedback and insight in the development of this Guide.

Any errors of fact or understanding that remain are solely the responsibility of the author.

Prepared by Raym Crow
Chain Bridge Group

1.703.536.7447 ▪ crow@chainbridgegroup.com ▪ www.chainbridgegroup.com

A Guide to Institutional Repository Software

CONTENTS

1) INTRODUCTION

1.1 Document Purpose

Universities and research centers throughout the world are actively planning and implementing institutional repositories. This activity entails policy, legal, educational, cultural, and technical components, most of which are interrelated and each of which must be satisfactorily addressed for the repository to succeed.

The Open Society Institute intends this guide to help organizations with one facet of their repository planning: selecting the software system that best satisfies their institution's needs. These needs will be driven by each institution's content policies and by the various administrative and technical procedures required to implement those policies. Therefore, this guide is designed for institutions already familiar with the various administrative, policy, and related planning issues relevant to implementing an institutional repository. Organizations just starting their evaluation of the benefits and features offered by an institutional repository should first refer to the growing background literature as a context for using this guide.[1]

1.2 Document Scope

The software systems discussed here satisfy three criteria:

- They are available via an Open Source license—that is, they are available for free and can be freely modified, upgraded, and redistributed.[2]
- They comply with the latest version of the Open Archives Initiative metadata harvesting protocols—this OAI compliance helps ensure that each implementation can participate in a global network of interoperable research repositories. And,
- They are currently released and publicly available—several new systems are currently being developed. As these systems become available for public release, we will revise this guide to include them.

The systems presented in this guide—Archimede, ARNO, CDSware, DSpace, Eprints, Fedora, i-Tor, MyCoRe, and OPUS—meet these criteria and allow an institution to implement a complete framework for an OAI-compliant repository without resorting to in-house technical development. While this guide describes these solutions, it does not attempt to identify the "best" system or to recommend one system over another. In each institution's case, the best software will be that which aligns well with the institution's particular requirements.

The System Description section has two parts: 1) a summary description of each system (Section 2.1) which provides a brief overview, contact information, and links for further information; and 2) a Feature & Functionality Table (Section 2.2) which provides additional detail on specific system functionality.

[1] The SPARC institutional repository information page points to a variety of such resources. See: <http://www.arl.org/sparc/core/index.asp?page=m0>.

[2] Of the systems described here, only ARNO requires a proprietary software component (Oracle). However, for some of the systems, use of proprietary software as a database management system (for example, Oracle or DB2) and/or operating systems (for example, Windows, Solaris) is optional.

The software systems described here were developed with various design philosophies and goals. The summary descriptions of the software in Section 2.1 provide overviews of the design philosophy for each system and offer some indication of the types of implementations for which the software would be best suited. The System Feature & Functionality Table in Section 2.2 attempts to provide an evaluative framework that equitably compares the capabilities of these disparate systems. However, the inclusion of a feature in Section 2.2 does not indicate that the functionality is an essential feature for an institutional repository. The importance of a particular feature must be considered in the context of the system's overall design and the individual institution's local requirements.

This guide can only provide an overview of the available software. Further, these systems are evolving rapidly. Readers should also refer to the additional information on system features and functionality available directly from the software providers themselves. Links to this information are provided with each system description.

2) SYSTEM DESCRIPTIONS

2.1 Summary System Descriptions

Archimede

Developed by Laval University Library in Quebec City, Canada, the Archimede project was designed to accommodate electronic preprints and post-prints from the institution's faculty and research staff. The Archimede institutional repository system complements two system components previously released by Laval. The first manages the university's electronic theses and dissertations; the second provides a production platform for electronic journals and monographs.

Archimede organizes the content submission process around a network of locally-managed research communities. Archimede was specifically designed to support multilingual international implementations. The text for the system's user interface is independent of the software code, facilitating the development of an interface in the local language. Archimede uses UTF-8 encoding and thus can accommodate any language. English, French, and Spanish language user interfaces are already implemented.

Archimede uses an indexing process, developed at Laval University, that integrates in a single occurrence two types of documents: a) a Dublin Core metadata record in XML; and b) the full text of the document(s) described by the metadata. These documents can be of any type, including HTML, PDF, MS Word, MS Excel, TXT, RTF, and others. Archimede supports the import and export of multiple types of metadata based on XSLT transformations.

Developed on a variety of Java Open Source technologies, Archimede runs on many operating systems (Windows, Linux, etc.) and can be used with several types of relational databases compatible with JDBC. This allows an institution flexibility in installing the software on an existing technical infrastructure.

Archimede Contact Information

Rida Benjelloun
Chef of Digital Development Section
Project Coordinator and Supervisor
Laval University Library
Pavillon Jean-Charles-Bonenfant
Laval University
Québec, Canada
G1K 7P4
rida.benjelloun@bibl.ulaval.ca
+418-656-2131 ext. 2090
http://archimede.bibl.ulaval.ca/

Additional Archimede Information

- An Archimede mailing list is available via Archimede@bibl.ulaval.ca

Archimede Software will soon be available on SourceForge.

ARNO

The ARNO project—Academic Research in the Netherlands Online—has developed software to support the implementation of institutional repositories and link them to distributed repositories worldwide (as well as to the Dutch national information infrastructure). The project is funded by IWI (Dutch acronym for "Innovation in Scientific Information Supply"). Project participants include the University of Amsterdam, Tilburg University, and the University of Twente. Released for public use in December 2003, the ARNO system has been in use at the universities of Amsterdam, Maastricht, Rotterdam, Tilburg, and Twente.

ARNO has different design goals from the other repository systems described here. It is designed to provide a flexible tool for creating, managing, and exposing OAI-compliant archives and repositories. The system supports the centralized creation and administration of repository content, as well as end-user submission.

The OAI-PMH module is not limited to presenting metadata in the standard (qualified) Dublin Core format, but offers a transformation engine that, based on the internal ARNO XML structures and XSLT style sheets, is able to produce any format. Other ARNO system features include: the ability to store versions of files; the ability to manage series (for example, of preprints or working papers and to set embargoes; and an interface to LDAP.

While ARNO offers considerable flexibility as a content management tool, it does not provide a self-contained, "off-the-shelf" institutional repository system. Following the toolbox approach, the ARNO system does not provide an end-user interface with end-user search capabilities. To be able to offer these services ARNO implementers need to deploy third party software (e.g. iPort or i-Tor).

Beyond the system functionality required to support institutional repositories, the ARNO infrastructure, and its simple and flexible data model, has the potential to interface easily with other third-party systems.

ARNO Contact Information

Thomas W. Place
Tilburg University
PO Box 90153, 5000 LE Tilburg
The Netherlands
+31 13 466 2474
T.W.Place@uvt.nl
http://www.uba.uva.nl/arno

Additional ARNO Information

- Vos, Fred. Presentation on ARNO at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/ fullAgenda.php?ida=a035925>

ARNO Software Available from <http://arno.uvt.nl/~arno/arnodist/>

CERN Document Server Software (CDSware)

The CERN Document Server Software (CDSware) was developed to support the CERN Document Server. The software is maintained and made publicly available by CERN (the European Organization for Nuclear Research) and supports electronic preprint servers, online library catalogs, and other web-based document depository systems. CERN uses CDSware to manage over 350 collections of data, comprising over 550,000 bibliographic records and 220,000 full-text documents, including preprints, journal articles, books, and photographs.

CDSware was designed to accommodate the content submission, quality control, and dissemination requirements of multiple research units. Therefore, the system supports multiple workflow processes and multiple collections within a community. The service also includes customization features, including private and public baskets or folders and personalized email alerts.

CDSware was built to handle very large repositories holding disparate types of materials, including multimedia content catalogs, museum object descriptions, and confidential and public sets of documents. Each release is tested live under the rigors of the CERN environment before being publicly released.

CDSware Contact Information

Jean-Yves Le Meur
CERN
CH-1211
Geneva, Switzerland
jean-yves.le.meur@cern.ch
+41-22-7674745
http://cdsware.cern.ch

Additional CDSware Information

- Le Meur, Jean-Yves. Presentation on CDSWare at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/fullAgenda.php?ida=a035925>

- Vesely, Martin and Thomas Baron, Jean-Yves Le Meur *et al*. "CERN Document Server: Document Management System for Grey Literature in a Networked Environment." *Publishing Research Quarterly* 20/1 (2004): 77-83.

There are two CDSware-related mailing lists:

- *project-cdsware-announce@cern.ch*
  Available at <http://cdsware.cern.ch/lists/project-cdsware-announce/archive/>

  Moderated, low-volume, read-only mailing list to announce new CDSware releases and other major news concerning the project.

- *project-cdsware-users@cern.ch*
  Available at <http://cdsware.cern.ch/lists/project-cdsware-users/archive/>

  Unmoderated, potentially high-volume mailing list, intended for discussion among users and developers of CDSware.

CDSWare Software Available from <http://cdsware.cern.ch/download/>

DSpace

MIT's DSpace was expressly created as a digital repository to capture the intellectual output of multidisciplinary research organizations. MIT designed the system in collaboration with the Hewlett-Packard Company between March 2000 and November 2002. Version 1.2 of the software was released in April 2004. The system is running as a production service at MIT, and a federation comprising large research institutions is in development for adopters worldwide.

DSpace integrates a user community orientation into the system's structure. This design supports the participation of the schools, departments, research centers, and other units typical of a large research institution. As the requirements of these communities might vary, DSpace allows the workflow and other policy-related aspects of the system to be customized to serve the content, authorization, and intellectual property issues of each.

Supporting this type of distributed content administration, coupled with integrated tools to support digital preservation planning, makes DSpace well suited to the realities of managing a repository in a large institutional setting.

DSpace is also focused on the problem of long-term preservation of deposited research material. Some of the system's adopters are actively engaged in research and development in this area. Over time, this should allow DSpace adopters to offer services both for hosting institutional repository content and maintaining the content for archival time frames.

DSpace Contact Information

MacKenzie Smith
Associate Director for Technology
MIT Libraries
Building 14S-208
77 Massachusetts Avenue
Cambridge, MA USA 02139
kenzie@mit.edu
(617) 253-8184
http://www.dspace.org/

Additional DSpace Information

- Bass, Michael J. et al. DSpace: Internal Reference Specification: Technology and Architecture. Version 2002-03-01 (2002).
  Available at <http://dspace.org/technology/architecture.pdf>

- Jones, Richard. "DSpace vs. ETD-db: Choosing software to manage electronic theses and dissertations." *Ariadne* 38 (January 2004).
  Available at: <http://www.ariadne.ac.uk/issue38/jones/intro.html>

- Morgan, Peter and William Nixon. Presentation on DSpace at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/fullAgenda.php?ida=a035925>

- Nixon, William. "DAEDALUS: Initial Experiences with EPrints and DSpace at the University of Glasgow." *Ariadne* 37 (October 2003). Available at <http://www.ariadne.ac.uk/issue37/nixon/>

  Article recounts the experience of the University of Glasgow in setting up an institutional repository using the DSpace software.

- Smith, MacKenzie, Mary Barton, Mick Bass, Margret Branschofsky, Greg McClellan, Dave Stuve, Robert Tansley, and Julie Harford Walker. "DSpace: An Open Source Dynamic Digital Repository." *D-Lib Magazine* 9 (January 2003).
  Available at <http://www.dlib.org/dlib/january03/smith/01smith.html>

  Describes the DSpace system, including its functionality and its design approach to addressing various issues in repository implementation. Also discusses MIT's implementation of DSpace.

DSpace Software Available from <http://www.dspace.org/resource/start.html>

Eprints

The Eprints software has the largest—and most broadly distributed—installed base of any of the repository software systems described here. Developed at the University of Southampton,[3] the first version of the system was publicly released in late 2000. The project was originally sponsored by CogPrints, but is now supported by JISC, as part of the Open Citation Project, and by NSF.

Eprints' worldwide installed base affords an extensive support network for new implementations. The size of the installed base for Eprints suggests that an institution can get it up and running relatively quickly and with a minimum of technical expertise. The number of Eprints installations that have augmented the system's baseline capabilities—for example, by integrating advanced search, extended metadata, and other features—indicates that the system can be readily modified to meet local requirements.

Eprints.org Contact Information

Christopher Gutteridge
Department of Electronics and Computer Science
University of Southampton
SO17 1BJ
United Kingdom
cjg@ecs.soton.ac.uk
+44 (0)23 8059 4833
http://software.eprints.org/

Additional Eprints.org Information

- Carr, Leslie. *EPrints Handbook*. Available at <http://software.eprints.org/handbook/>

  Provides guidance to researchers and research managers in the adoption of an Eprints server.

- Gutteridge, Christopher. Presentation on GNU Eprints at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/fullAgenda.php?ida=a035925>

- Nixon, William J. "The evolution of an institutional e-prints archive at the University of Glasgow." *Ariadne* 32 (June-July 2002). Available at <http://www.ariadne.ac.uk/issue32/eprint-archives/intro.html>

- ________. "DAEDALUS: Initial Experiences with EPrints and DSpace at the University of Glasgow." *Ariadne* 37 (October 2003). Available at <http://www.ariadne.ac.uk/issue37/nixon/>

  Articles recount the experiences of the University of Glasgow in setting up an institutional repository using the Eprints.org software.

[3] Eprints was written by Rob Tansley (based on the CogPrints software, which was written by Matt Hemus), and subsequently upgraded and maintained by Christopher Gutteridge.

- Pinfield, Stephen, Gardner, Mike and MacColl, John. "Setting up an institutional e-print archive." *Ariadne* 31 (March-April 2002).
  Available at <http://www.ariadne.ac.uk/issue31/eprint-archives/>

  Article describes the main issues involved with establishing an institutional repository and discusses some of the practical issues that arise in the initial stages of implementing an Eprints.org repository.

- Sponsler, Ed and Eric F. Van de Velde. "Eprints.org Software: A Review." *SPARC eNews* (August-September 2001).
  Available at <http://www.arl.org/sparc>

  An early review of the Eprints.org software and comments on an initial repository implementation at the California Institute of Technology.

- Discussion forum for Eprints users: <http://community.eprints.org/phpBB/>

Eprints Software Available from <http://software.eprints.org/download.php>

Fedora

The Fedora digital object repository management system is based on the Flexible Extensible Digital Object and Repository Architecture (Fedora). The system is designed to be a foundation upon which full-featured institutional repositories and other interoperable web-based digital libraries can be built.

Jointly developed by the University of Virginia and Cornell University, the system implements the Fedora architecture, adding utilities that facilitate repository management. The current version of the software provides a repository that can handle one million objects efficiently. Subsequent versions of the software will add functionality important for institutional repository implementations, such as policy enforcement, versioning of objects, and performance enhancement to support very large repositories.

The system's interface comprises three web-based services:

- A management API that defines an interface for administering the repository, including operations necessary for clients to create and maintain digital objects;
- An access API that facilitates the discovery and dissemination of objects in the repository; and
- A streamlined version of the access system implemented as an HTTP-enabled web service.

Fedora supports repositories that range in complexity from simple implementations that use the service's "out-of-the-box" defaults to highly customized and full-featured distributed digital repositories.

Fedora Contact Information

Ronda Grizzle
Technical Coordinator, Fedora Project
Digital Library Research & Development
University of Virginia
Charlottesville, VA USA 22903
rgrizzle@virginia.edu
http://www.fedora.info/

Additional Fedora Information

- Jantz, Ronald. "Public Opinion Polls and Digital Preservation: An Application of the Fedora Digital Object Repository System." *D-Lib Magazine* 9/11 (2003). Available at <http://www.dlib.org/dlib/november03/jantz/11jantz.html>

- *Mellon Fedora Technical Specification* (December 2002). Available at <http://www.fedora.info/documents/master-spec-12.20.02.pdf>

- Payette, Sandy. Presentation on Fedora at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/ fullAgenda.php?ida=a035925>

- Staples, Thornton, Ross Wayland, and Sandra Payette. "The Fedora Project: An Open Source Digital Object Management System." *D-Lib Magazine* 9/4 (April 2003). Available at <http://www.dlib.org/dlib/april03/staples/04staples.html>

- Additional articles and papers available from <http://www.fedora.info/pubs.shtml#pubs>

Fedora Software Available from <http://www.fedora.info/release/1.2.1/>

i-Tor

i-Tor—Tools and technologies for Open Repositories—was developed by the Innovative Technology-*Applied* (IT-*A*) section of Netherlands Institute for Scientific Information Services (Dutch acronym: NIWI).[4] i-Tor development concentrates on four areas: e-publishing; repositories; the content management system; and "collaboratories." NIWI offers i-TOR as a web-based technology by which users can present various types of information through a web interface, irrespective of where the data is stored or the format in which it is stored. i-Tor aims to implement a "data independent" repository, where the content and the user-interface function as two independent parts of the system. In essence, i-Tor acts as both an OAI service provider, able to harvest OAI compatible repositories and other databases, and an OAI data provider.

Because i-Tor is able to publish data from a variety of relational databases, file systems, and websites, the system allows an institution considerable latitude in the way it organizes its repository. It can create new databases for the repository, but it can also use already existing relational databases. Further, i-Tor supports harvesting of data directly from a researcher's personal home page. The system's design allows an end user to add content via a web browser without a software developer acting as an intermediary.

[4] See: <www.niwi.knaw.nl>.

Because of this design, i-Tor does not enforce a specific workflow on a group or subgroup. Rather, i-Tor gives an institution tools (for example, fine grained security, notification, etc.) to set up any required workflow required by the organization, without integrating this workflow into the i-Tor system itself. i-Tor's design might make it an appropriate choice for an institution that wishes to impose a repository on top of an existing set of disparate digital repositories.

i-Tor Contact Information

Henk Harmsen
Head of Operational Management
Netherlands Institute for Scientific Information Services
henk.harmsen@niwi.knaw.nl
+31 20 462 8605
http://www.i-tor.org/en/toon

Additional i-Tor Information

- NIWI's virtual lab for Open Standards and Open Software <http://www.i-tor.org>

- Reposi-Tor: i-Tor's electronic newsletter. Available at <http://www.itor.org/en/reposi_tor_i_tors_elektronische_nieuwsbrief/>

- Sesink, Laurents. Presentation on i-Tor at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/fullAgenda.php?ida=a035925>

i-Tor Software Available from <http://sourceforge.net/projects/i-tor/>

MyCoRe

MyCoRe grew out of the MILESS Project of the University of Essen. The MyCoRe system is now being developed by a consortium of universities to provide a core bundle of software tools to support digital libraries and archiving solutions (or Content Repositories, hence "CoRe"). The bundle is designed to be configurable and adaptable to local requirements (hence, "My"), without the need for local programming efforts.

In contrast to MILESS, which provides a hard-coded Qualified Dublin Core data model, the MyCoRe data model is completely configurable. Further, MyCoRe provides a sample application, based upon a "core" of functionality, that shows users how to build their own applications using metadata configuration files. The core contains all the functionality that would be required in a repository implementation, including distributed search over geographically dispersed MyCoRe repositories, OAI functionality, integrated audio/video streaming support, file management, and online metadata editors. Local implementations can customize the core to serve their particular requirements.

MyCoRe is not hard-coded to a particular underlying database. Rather, a persistence layer interface is provided, together with implementations for different databases. In addition to implementations for multiple Open Source database systems, there is also support for the commercial IBM Content Manager system, which can be used for very large repositories.

MyCoRe Contact Information

Frank Lützenkirchen
Technical Contact
Essen University Library
University of Duisburg-Essen
Universitätsstraße 9-11
45141 Essen, Germany
luetzenkirchen@bibl.uni-essen.de
+49-(0)201-183-2124
http://www.mycore.de

Additional MyCoRe Information

- Lützenkirchen, Frank. Presentation on MyCoRe at the Third CERN Workshop on Innovations in Scholarly Communication: Implementing the benefits of OAI. February 12, 2004. CERN, Geneva, Switzerland. Available at <http://agenda.cern.ch/fullAgenda.php?ida=a035925>

MyCoRe Software Available from <http://www.mycore.de/engl/index.html>

OPUS

OPUS—Online Publications of the University of Stuttgart[5]—was developed in 1998 by the University Library and the Computing Center of the University of Stuttgart. The goal of the original project was to provide a system by which faculty, students, and staff at the university could manage their electronic publications, including published and unpublished articles and theses and dissertations.

The OPUS software is currently used by about thirty-five other German universities to manage the electronic publications of their university populations, and the system supports a search of metadata at participating German institutions (not all of which are using OPUS as their repository platform).[6] Most OPUS implementations are managed and operated by an institution's university library, although some represent cooperative efforts of the library and the university's press and/or academic computing center.[7] OPUS is also being used by at least one discipline-specific repository.[8]

The initial development project, funded by the German Research Net and the German Federal Department of Higher Education, ended in October 1998. Ongoing development of OPUS is now funded by the University of Stuttgart. Main features for future development include digital signatures and multimedia documents.

[5] For implementations other than at the University of Stuttgart, the OPUS acronym is sometimes rendered **O**nline **Pu**blications **S**ystem.

[6] See <http://elib.uni-stuttgart.de/opus/gemeinsame_suche.php>.

[7] See, for example, the cooperative implementation at the University of Weimar, at <http://e-pub.uni-weimar.de/>.

[8] A pan-institutional digital repository for research in psychology is run jointly by Saarbruecken library and ZPID, the institute for psychology information for the German-speaking countries at Trier University. See: <http://psydok.sulb.uni-saarland.de/>.

The OPUS interfaces and documentation are primarily in German, and all current implementations of the software are in Germany. Therefore, the system would appear to have its most direct appeal to repository implementations in German-speaking countries.

OPUS Contact Information

Frank Scholze
Subject Specialist and Head, Public Services Department
Stuttgart University Library
University of Stuttgart
Holzgartenstrasse 16, D-70174, Stuttgart, Germany
scholze@ub.uni-stuttgart.de
+49 (0)711/121-2269

Or:

Annette Maile
Stuttgart University Library
University of Stuttgart
Holzgartenstrasse 16, D-70174, Stuttgart, Germany
maile@ub.uni-stuttgart.de
+49 (0)711/121-4189
http://elib.uni-stuttgart.de/opus/doku/english/index_english.php

Additional OPUS Information

- Hauser, Jürgen, Frank Scholze and Uwe Albrecht. "MAVA - Entwicklung und Integration eines erweiterbaren multimedialen Dokumentensystems." In Rützel-Banz, Margit (Hrsg.), *Grenzenlos in die Zukunft*: 89. Deutscher Bibliothekartag in Freiburg im Breisgau. Frankfurt: Klostermann, 2000, 57 - 69.

- Maile, Annette and Frank Scholze. "Online Publikationsverbund der Universität Stuttgart (OPUS)" In BI - *Informationen für Nutzer des Rechenzentrums* 11/12 (1997): 21-24.

- __________. "Online Publikationsverbund der Universität Stuttgart (OPUS)" In *Stuttgarter Unikurier* 77/78 (February 1998): 12

- Scholze, Frank. "Einbindung elektronischer Hochschulschriften in den Verbundkontext am Beispiel OPUS." In Tröger, Beate (Hrsg.), *Wissenschaft online: Elektronisches Publizieren in Bibliothek und Hochschule*. Frankfurt: Klostermann, 2000, 406 - 420.

- Stephan, Werner and Frank Scholze. "Online Publikationsverbund: Erfassung und Organisation elektronischer Hochschulschriften" In *Bibliotheksdienst* Heft 1 (1999): 92-102

- Additional articles and papers available from <http://elib.uni- stuttgart.de/opus/doku/veroeff.php>

OPUS Software Available from <http://elib.uni-stuttgart.de/opus/doku/english/index_english.php>

Summary

As noted in the introduction, each of the systems above derives from a design philosophy that reflects the original requirements of the developing institution(s). Archimede was specifically designed to support multilingual implementations; ARNO provides a system for the centralized management of metadata; CDSware handles very large repositories accommodating disparate types of materials; DSpace supports community-based content policies and submission processes, and provides tools to support the preservation of the digital objects submitted; Eprints supplies a straightforward and useful repository system, with a large and active installed user community; Fedora provides a full-featured digital library system that can accommodate very large repositories; i-Tor offers a toolkit for constructing an environment in which the contents of multiple databases can be accessed and displayed in an integrated manner; MyCoRe stresses flexibility and the ability to configure the software to support disparate digital libraries and repository databases; and OPUS offers a large and varied installed user base in Germany. Again, the local requirements of each repository implementation will dictate which system will best serve an institution's needs.

2.2 Feature & Functionality Table

Feature	Archimede	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe	OPUS
Technical Specifications									
1.0 Standards Information									
1.1 OAI-PMH version supported	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0	OAI-PMH 2.0
1.2 Z39.50 protocol compliant	No	No	No	No	No	No	No	No[1]	No
1.3 Open source license [1]	GNU GPL	TBD	GNU GPL	BSD	GNU GPL	MPL	GNU GPL	GNU GPL	BSD[1]
1.4 Latest version release date	May-04	Dec-03	Aug-02	Apr-04	Mar-02	Apr-04	Apr-04	Oct 03	Nov-03
1.5 Latest version number	1.0	1.0	0.0.9	1.2	2.3.6	1.2.1	niwi-2004-04-19	0.9	2.0
2.0 Hardware									
2.1 Minimum hardware requirements [2]	No specific requirements	No specific requirements	No specific requirements[1]	No specific requirements[1]	No specific requirements	No specific requirements	No specific requirements	No specific requirements[2]	No specific requirements
2.2 SAN support[3]	No	No	No	Yes	Yes	Yes	Yes	No	No
3.0 Software									
3.1 Operating system (tested)	Linux/Windows	Linux/Solaris	Linux/Solaris	UNIX/MacOSX/ Windows[2]	GNU/Linux/Solaris[1]	Unix/MacOSX/Windows[1]	Linux/Windows	AIX/Windows/Linux/ Solaris	Linux/Solaris/AIX/IRIX
3.2 Programming language	Java	Perl	Python/PHP	Java	Perl	Java	Java	Java	PHP
3.3 Database	Many[1]	Oracle 8i[1]	MySQL	PostgreSQL/Oracle[3]	MySQL	MySQL/McKoi/Oracle[2]	MySQL, Oracle, SQL Server, Berkeley database	MySQL, PostgreSQL; XML:DB compliant; Commercial databases[3]	MySQL[2]
3.4 Web server	Any	Apache	Apache/PHP, Python	Any[4]	Apache 1.3 [2]	Tomcat 4.1	Jetty	Apache	Any
3.5 Java servlet engine	Any[2]	N/A	N/A	Any[4]	N/A	Tomcat 4.1	Jetty	Any[4]	N/A
3.6 Search engine	Lucene	N/A[2]	cdsware[2]	Lucene	N/A	Database[3]	Lucene	Via JDBC and XML:DB	htDig[3]
3.7 Other	Lius, OAICat, Torque, Struts	N/A	WML: Website META Language	OAICat, SRW	N/A	N/A	N/A	Apache Ant build tool	N/A
4.0 Clients supported	Any browser with minimal CSS & Javascript support	Any browser with minimal CSS & Javascript support	All HTML 4.0 clients	All web browsers	Netscape, Mozilla, IE, Lynx[3]	Web browsers and SOAP clients	Netscape, Mozilla, IE	All web browsers	All web browsers
5.0 Staff requirements[4]									
5.1 UNIX systems administrator	For setup[3]	Yes[3]	Yes	Yes	Yes	For setup[4]	Recommended [1]	Recommended	Yes
5.2 Java programmer	Recommended	No	No	Recommended	No	Recommended	No	Recommended[5]	No
5.3 PERL programmer	No	Recommended	No	No	Recommended[4]	No	No	No	No[4]
5.4 Python programmer	No	No	No[3]	No	No	No	No	No	No
6.0 Installed base									
6.1 Number of installations	1	7	7+[4]	20+[5]	140 [5]	20 [5]	30	10 [6]	37
6.2 Geographic coverage	Canada	Netherlands	Europe & US[5]	Worldwide	Worldwide[6]	Worldwide[6]	Netherlands	Germany & Sweden	Germany

Feature	Archimede	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe	OPUS
Repository & System Administration									
7.0 Set-up/Installation									
7.1 Automated installation script	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[5]
7.2 System update script	Yes	Yes	Yes	Yes[6]	Yes[7]	Yes	Yes	Via CVS repository	Yes[5]
7.3 Update system update without overwriting customized features[5]	Yes	Yes[4]	Yes	Yes	Yes[8]	Yes	Yes	Yes[7]	Yes
8.0 Module-level API(s)[6]	Yes[4]	No	Yes[6]	Yes[7]	Yes	Yes[7]	Yes[2]	Yes	No
9.0 User registration, authentication & password administration									
9.1 Password administration	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.1.1 System-assigned passwords	Yes	No	Yes[7]	Yes	No	No	No	No	Yes[6]
9.1.2 User selected passwords	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.1.3 Forgotten password function [7]	Yes	Yes[5]	Yes	Yes	Yes	No	No	No	No
9.2 User registration verification/Other security mechanisms [8]	Database table	LDAP and/or ARNO registry	MySQL table/Apache ACL	email/X.509	MySQL table[9]	No	LDAP, A-Select[3]	RDBMS table	No
9.2.1 Edit user profile	Yes	No	Yes	Yes	Yes	No	Yes		No
9.3 Limit Access by User Type[9]	Yes	Yes	Yes	Yes	Yes	Yes[8]	Yes	Yes	No
9.4 Multiple Authentication Methods[10]	No	Yes	Yes	Yes	No	Yes[9]	No[3]	Planned	No[7]
9.5 Limit Access at File/Object Level[11]	Yes	Yes	Yes	Yes	Yes	No[10]	Yes	No	No
10.0 Content Submission Administration									
10.1 Define multiple collections within same instance of system[12]	Yes	Yes	Yes[8]	Yes	Yes	Yes	Yes	Yes	Yes
10.1.1 Set different submission parameters for each collection[13]	No	Yes	Yes	Yes	No	Yes[11]	Yes	No	No
10.1.2 Home page for each collection	Yes	No	Yes[9]	Yes	No	Yes	Yes	No	Yes
10.2 Submission Stages[14]	Pending, Approved		Submit, Modify, Revise, Approve, etc.[10]	Assemble, Pending, Approved		Ingest, Create, Modify, Activate, Deactivate	Yes[4]	No[1]	Submit, Revise, Approve
10.2.1 Segregated submission workspace [15]	Yes	Yes	Yes	Yes	Yes[10]	Yes	Yes[4]	No	No[7]
10.2.2 Submission roles[16]	Administrator, Community Administrators, User	Contributors, Editors, Administrators, Site Managers	Submitters, Moderators, Reviewers, Approvers, Administrators	Submitters, Reviewers, Approvers, Editors	User, Editor, Administrator[11]	Administrator	Yes[4]	No	User, Administrators
10.2.3 Configurable submission roles within collections[17]	Yes	Yes	Yes	Yes		No	Yes[4]	No	No
10.3 Submission Support									
10.3.1 Email notification for submitters[18]	Yes	Only during registration	Yes[9]	Yes	Yes	No	Yes	No	Yes
10.3.2 Email notification for content administrators[19]	Yes	Yes	Yes[9]	Yes	Yes	No	Yes	No	Yes
10.3.3 Personalized system access for registered users[20]	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
10.3.3.1 View pending content submissions [21]	Yes	Yes	Yes	Yes	Yes	No	Yes[4]	No	No
10.3.3.2 View approved content [22]	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
10.3.3.3 View pending content administration tasks [23]	Yes	Yes	Yes	Yes		No	Yes[4]	No	Yes

Feature	Archimede	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe	OPUS
10.3.4 Distribution license[24]									
10.3.4.1 Request distribution license [25]	No	No	No	Yes	No	Yes[12]	No	No	Yes
10.3.4.2 Store distribution license with content [26]	No	No	No	Yes	No[12]	Yes	No	No	No
11.0 System generated usage statistics and reports									
11.1 System-generated usage statistics[27]	No	Yes	No[11]	Yes	No[13]	Yes[13]	Yes[5]	No	Yes
11.2 Usage reports[28]	No	No	No	Yes	No	No[14]	Yes	No	Yes[8]
Content Management									
12.0 Content Import/Export									
12.1 Upload compressed files	Yes	Yes	Yes	Yes[8]	Yes	Yes	Yes	No[1]	Yes
12.2 Upload from existing URL	No	Yes	Yes	No	Yes	Yes	Yes[6]	No[1]	No
12.3 Volume import for objects[29]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[9]
12.4 Volume import for metadata[30]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[9]
12.5 Volume export/content portability[31]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
13.0 Document/Object Formats									
13.1 Approved file format function[32]	No[5]	Yes	Yes	Yes	Yes	No[15]	Yes[4]	No	Yes
13.2 File formats ingested[33]	All	All	All[12]	All	All[14]	All	All	All	All
13.3 Submitted items can comprise multiple files[34]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
14.0 Metadata									
14.1 Metadata schema supported [35]	Qualified Dublin Core	Dublin Core	Standard Marc21	Qualified Dublin Core	Dublin Core	Dublin Core	Any	Qualified Dublin Core[8]	Qualified Dublin Core
14.2 Support for extended metadata[36]	No	Yes	Yes	Custom	Yes	Yes	Any	Any[9]	Yes
14.3 Metadata review support[37]	Yes	Yes	Yes	Yes	Accept, Edit, Bounce (require changes), Delete	No	Yes[4]	No	Yes
14.4 Metadata export[38]	Yes	Yes	OAI-Marc export	METS & Custom XML Schema	Custom XML Schema	Yes	Yes	Yes	Yes
14.5 Disallow metadata harvesting[39]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
14.6 Add/delete metadata fields	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
14.7 Set default values for metadata [40]	No	Yes	Yes	Yes		No	Yes		Yes
14.8 Supports Unicode character set for metadata	Yes	Partial[6]	Yes	Yes	Yes	Yes	Yes	Yes	No
15.0 Real-time updating and indexing of accepted content	Yes	No	Yes	Yes	Yes[15]	Yes	Yes	Yes	Yes[10]

Feature	Archimede	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe	OPUS
Dissemination (User Interface & Search Functionality)									
16.0 User Interface									
16.1 Modify interface "look & feel"[41]	Yes	Yes[7]	Yes	Yes[10]	Yes[16]	Yes	Yes	Yes	Yes
16.2 Apply a custom header/footer to static or dynamic pages	Yes	No	Yes[13]	No	Yes	Yes	Yes	Yes	Yes
16.3 Supports multiple language interfaces	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
16.4 End user document folders[42]	Yes	No	Yes	No	No	No	Yes		No
16.5 Discussion forum support[43]	No	No	No[14]	No	Yes[17]	No	Yes	No	No
17.0 Search Capability									
17.1 Full text[44]	Yes[6]	No[2]	Yes	Yes[11]	No[18]	No	Yes	Yes	Yes
17.1.1 Boolean logic	Yes	No	Yes	No	No	No	Yes	No	Yes
17.1.2 Truncation/wildcards[45]	Yes	No	Yes	No	No	No	Yes	Yes	No
17.1.3 Word stemming[46]	Yes	No	No	No	No[19]	No	No	Yes	Yes
17.2 Search all descriptive metadata[47]	Yes	No	Yes	Yes	Yes	Yes[16]	Yes	Yes	Yes
17.2.1 Boolean logic	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes
17.2.2 Truncation/wildcards	Yes	No	Yes	Yes	No	Yes	Yes		Yes
17.2.3 Word stemming	Yes	No	No	Yes	No	No	Yes	Yes	No
17.3 Search selected metadata fields[48]	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
17.4 Browse									
17.4.1 By author	Yes	No	Yes	Yes	Yes[20]	Yes[17]	Yes[7]	Yes	No[11]
17.4.2 By title	Yes	No	Yes	Yes	Yes[20]	Yes	Yes[7]	Yes	No
17.4.3 By issue date	Yes	No	Yes	Yes	Yes[20]	Yes	Yes[7]	Yes	No
17.4.4 By subject term	Yes	No	Yes	No	Yes[20]	Yes	Yes[7]	Yes	Yes
17.4.5 By collection	Yes	No	Yes	Yes	Yes[20]	Yes	Yes[7]	Yes	Yes
17.5 Sort search results									
17.5.1 By author	No	No	Yes	No	Yes	No	Yes	Yes	No
17.5.2 By title	No	No	Yes	No	Yes	No	Yes	Yes	Yes
17.5.3 By issue date	No	No	Yes	No	Yes	No	Yes	Yes	Yes
17.5.4 By relevance	Yes	No	No	No	No	No	Yes		Yes[12]
17.5.5 By other	No	No	Any metadata field	No	Yes[21]	No	Yes[7]	Yes	No
18.0 Indexed by Google/Other Search Engines[49]	Possible	Possible[8]	Possible[15]	Yes	Possible	Possible[18]	Yes	Possible	Yes
Archiving									
19.0 Persistent document identification[50]									
19.1 System-assigned identifiers	Yes	Yes	Yes	Yes	Yes	Yes[19]	Yes	Yes	Yes
19.2 CNRI Handles[51]	No	No	No	Yes	No	No	No	No[10]	No[13]
20.0 Data preservation support									
20.1 Defined digital preservation strategy[52]	Yes	No[9]	Yes[16]	Yes	No	Yes	Yes	No[1]	Partial[14]
20.2 Preservation metadata support (see also 14.2)[53]	Yes	Yes	Yes[17]	Yes	No	Yes	No[3]	No[1]	Partial[14]
20.3 Data integrity checks	No	No	No	MD5 checksum	MD5 checksum	SIP schema validation	Yes	MD5 checksum	No
21.0 Object history/Version control	Versioning system	Versioning system for both metadata & objects	Versioning system	ABC Harmony data model	Some	Linear[20]	No[10]	No[1]	No

Feature	Archimede	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe	OPUS
System Maintenance									
22.0 System support									
22.1 Documentation/manual	Yes	Yes[10]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
22.2 Listserv	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
22.3 Bug track/feature request system	Yes[7]	No	Yes	Yes[12]	No	Yes	Yes[11]	No	No
22.4 Formal support/help desk	No	No	For fee	No	No[22]	Yes	No	No	No

NB: A blank cell in the table indicates insufficient information to provide a definitive response.

Notes on System Features & Functionality

1) For most of the systems discussed here, the operating system and all of the supporting software are Open Source software licensed under the GNU General Public License (GPL). MIT and Hewlett-Packard have agreed to license all DSpace software with an open source, BSD license, and DSpace intends to add any third-party components under the same terms. The Fedora repository system is open source software licensed under the Mozilla Public License.

2) Given the variety of local conditions, none of the systems specify minimum CPU requirements. Where the system web site describes potential hardware configurations, we have provided a link to that information.

3) Indicates that the system can operate on a storage area network (SAN).

4) Depending on the software indicated under Item 3.0 ("Software"), some systems will require some staff technical experience with the operating system, storage system, webserver, command manager, and/or search engine. Systems administrators and programmers can be allocated resources and not necessarily full-time staff, depending on the scale and requirements of a particular implementation.

5) Allows the system to be updated without overwriting the modifications an institution might make to page templates, emails, help pages, search pages, etc.

6) Most of the systems allow some level of local customization of the system. In some systems this is accomplished by modifying scripts. Others provide an Application Programmer Interface (API) that allows a programmer at the adopting institution to modify system functionality.

7) Provides a secure process by which users who have forgotten their passwords can select a new password without human intervention. Typically, the system uses the user's email address to administer the new password.

8) Registers and authenticates users who are authorized to submit content to and/or administer content in the repository, as distinct from the global audience of anonymous users who can access content that is publicly accessible.

9) Allows the repository administrator to limit access to certain content based on the user's level of authorization. This could be used, for example, to limit access to an academic department's working papers to faculty members in that department. Similarly, it could be used to limit access to materials that are restricted by research funding stipulations.

10) Allows the repository administrator to apply various levels of access restrictions to submitted items based on user type. For example, most items would be accessible globally to all users; some items might be available via IP address to a university community; and other items might be limited to ID/password access to a relatively small group of users.

11) Allows the repository system administrator to restrict access to individual files within an item submission. For example, a dissertation might contain images or other component files to which access should be restricted.

12) Allows the institution to define multiple content collections and/or groups of users within one installation of the system. Collections could be defined in various ways, including by subject matter, content type or purpose, audience, etc. (e.g., a working paper series or collection of curriculum support materials). User groups could represent academic departments, schools, research institutes, administrative departments (e.g., museums, hospitals, etc.), as needed to address the needs of the implementing institution.

13) Allows the repository administrator to set different content submission and review/approval parameters (if desired) for each of the collections and/or user groups defined within the repository.

14) Allows repository system administrators to designate the number and types of stages through which content might pass from initial submission to inclusion in the repository.

15) Provides a separate pre-public workspace that stores incomplete and/or pre-approval stage content submissions. This can simplify the process for submitting a document by allowing the user to save an interrupted or incomplete submission, rather than abandon an incomplete submission altogether.

16) Provides for a configurable set of review functions and administration within a repository. (For example, content approval (per whatever criteria the user group has adopted); metadata review, editing, and approval; etc.)

17) Some systems apply the same roles and process across all collections in the repository. Others specify these functions at the collection level, allowing different collections within one instance of the system to offer different submission and review processes.

18) Sends an email notification to a user regarding the status of a content submission (e.g., that the item has been approved for inclusion in the repository or has been returned to the submitter).

19) Sends an email notification to a content administrator (e.g., a reviewer, approver, etc.) when a submission has been routed to them for review, approval, etc.

20) Allows registered users access to content and process status information. This type of function can allows users to determine the status of content submissions and/or pending content approval tasks.

21) Allows users to review all the content that they have submitted to the repository.

22) Allows users to review and/or complete unfinished content submissions (that is, content submissions that were initiated, but not completed for some reason).

23) Allows content administrators (e.g., reviewers, editors, approvers, etc.) to review submissions awaiting processing.

24) To allow the host institution to administer and disseminate the material submitted to the repository, a repository typically needs each contributor to grant the institution an irrevocable, non-exclusive, royalty-free license to distribute the content, to translate its format for the purpose of digital preservation, and to maintain the content in perpetuity.

25) Allows the institution to integrate a request for rights to maintain and distribute the content as part of the content submission process. Some systems support multiple license terms, which may vary by content collection or by user. Others address such license terms by procedures outside the system software itself.

26) Allows the institution to store specific license terms with each content submission. As license terms may change over time, or by content type, this enforces clarity as to which terms apply to each submission.

27) Allows repository administrators to track the use and adoption of the repository. This facilitates system capacity planning and supports internal resource allocation and budget support issues.

28) Pre-set and/or configurable usage reports can add to the usefulness of system-generated usage statistics.

29) Allows an institution to import existing digital libraries and other digital material.

30) Allows a repository to import metadata for existing digital collections.

31) An explicit expectation for an institutional repository is that the content managed by the system will survive the system itself and can migrate as new technologies evolve. This feature refers to the manner in which content can be exported from the system.

32) This feature allows the system administrator to limit content submission to approved format types. This allows the repository to indicate which digital formats it is willing to accept (from a policy perspective) as opposed to which formats the system is capable of accommodating (from a technical perspective). This can help support repository policies designed to ensure ongoing access to, and preservation of, the repository’s contents.

33) Refers to the digital formats that a system is capable of ingesting (as opposed to those an institution may decide to support as a matter of policy).

34) Allows a user to submit multiple files and/or file types a part of a single deposit. This permits, for example, a user to submit a research paper along with its supporting data set or a conference paper along with the overhead presentation given at the conference.

35) This refers to the extent to which a system can store metadata related to a content submission and make that metadata searchable via a user interface. The OAI protocol harvests unqualified Dublin Core metadata, and all the systems described here support that baseline Dublin Core metadata, which is what makes it possible to search across repositories using the systems.

36) As a lowest common denominator, the unqualified Dublin Core will not be sufficiently detailed to serve the needs of many institutional repository collections. Therefore, in addition to the Dublin Core, the OAI protocol supports parallel metadata sets, allowing repositories to expose additional metadata specific to a particular collection or content type. Some systems support (or plan to support) other metadata standards, including those for domain-specific, preservation, and rights metadata.

37) For the metadata harvesting to be effective, a repository must establish a quality control process and quality threshold on the metadata stored in the system. This will prove especially true for repositories that intend to allow authors to self-archive their papers and provide their own metadata. This feature supports a metadata approval process whereby metadata can be reviewed, corrected, enhanced, and/or approved prior to being made available through the system.

38) Allows an institution to export the repository's metadata, in XML or some other structured format, to facilitate migration to a subsequent system.

39) Allows system administrator to "turn off" the ability of OAI harvesters to harvest metadata from the repository overall. This would effectively disable the repository's interoperability.

40) Allows the repository system administrator to establish defaults for metadata fields to simply metadata entry. For example, an institution field could be set to default to the hosting institution (for example, Institution="University of Pennsylvania").

41) Allows an institution to modify the look of the interface through an API or by adapting scripts that control the service's presentation.

42) Allows users to store repository content in personalized document folders within the system.

43) System supports discussion forums within the repository.

44) This item refers to the internal system search and retrieval software and presentation layer software, not to external service providers or search engines. Some of the systems that don't have an integrated search engine provide instructions for adding an Open Source search tool.

45) Allows the use of wildcards (for example, *=multiple characters; ?=single character).

46) Allows a search to return results based on the root form of a word. For example, “land” will also match “landed,” “landing,” lands,” and “landed.”

47) Allows a user to search all defined descriptive metadata fields.

48) Allows a user to search selected metadata fields. For example, search only the “title” or “author” fields.

49) Indicates that the system can be searched by Google and other internet search engines, if the search tool is pointed at the correct system server.

50) Persistent naming allows a repository to change its internal retrieval mechanisms and/or physically move content without compromising reference citations and other links. These persistent identifiers remain valid even were the repository content to be migrated to a new system or were management responsibility for the repository to be assigned to a third party.

51) The CNRI Handle System allows institutional repositories to achieve the continuity and persistent naming described above (see 20.0). The Handle System protocols enable a distributed computer system to store handles of digital resources and resolve those handles to locate and access the resources. The information associated with each handle can be changed to reflect the current state of the identified resource without changing the handle itself, thus allowing the name of the item, as well as reference citations and other links, to persist over changes of location and other state information.

52) Some systems have integrated features that facilitate the long-term digital preservation of submitted material. These can be important features, as preservation best practice suggests taking steps early in the life-cycle of an electronic resource mitigates the cost and technical difficulty of preserving it in the future. However, a successful digital preservation program also requires extensive policy development, funding, and planning to support such preservation support features. Further, it should not be inferred that absence of these features precludes digital preservation.

53) Preservation metadata stores technical information that supports preservation decisions and action, documents preservation action taken, records the effects of preservation strategies, to ensure the authenticity of digital resources over time, and notes information about collection management and the management of rights.

System-Specific Notes

Archimede Notes

1) Archimede is based on Apache DB Torque which supports a wide range of databases. For the complete list see <http://db.apache.org/torque/>.

2) Any Servlet 2.3+ compliant engine.

3) If server is run on Unix. Setup requires little OS-specific knowledge. Unix knowledge helpful for setting up init scripts, etc.

4) API and command line interface.

5) Planned.

6) Implemented by Lius which supports the following formats : HTML, XML, text, RTF, Word, Excel, and PDF.

7) Through the SourceForge system.

ARNO Notes

1) Port planned to PostgreSQL or other Open Source DBMS.

2) ARNO provides basic search functionality to support metadata maintenance, but relies on third-[arty search engines for end-user access.

3) Some XML/XSLT knowledge recommended. Additional metadata formats are exposed through the OAI-PMH interface by applying XSLT style sheets to the internal ARNO XML format.

4) Excluding changes in source code.

5) For users registered via LDAP.

6) Full support in development.

7) Some interface modification possible using CSS style sheets.

8) An HTML presentation of the metadata and links to the full text may be indexed. Options for making the full text available for indexing are under investigation.

9) Under development in conjunction with DARE project.

10) Partially completed; in development.

CDSware Notes

1) System requirements depend on collection size, number of expected users, database platform, etc.

2) CDSware uses its own indexing technology and search engine.

3) Only needed if institution intends to add new features to the system.

4) Exact number unknown as CERN does not follow up all installations/downloads of the CDSware package.

5) Switzerland (3), France, Germany, Italy, and the US.

6) API and command line interface.

7) Not mandatory.

8) Supports hierarchy of collections (any tree), as well as Virtual Collections ('horizontal views').

9) Configurable.

10) Wide range of options: see <http://doc.cern.ch/EDS/current/guide/english/>

11) Uses third-party tools, such as Webalizer.

12) CERN Conversion Server can be attached to CDSware to automate conversion to PDF (for documents): <http://doc.cern.ch/Convert>

13) The collections home page can also be customized.

14) In development for next release.

15) The HTML formats of CDSware records can either be created on-the-fly or they can be pre-processed, saved to files to allow web search engine indexing.

16) Automated conversion to PDF format.

17) Marc21 standard.

DSpace Notes

1) For suggested DSpace hardware configurations, see: http://dspace.org/what/dspace-hp-hw.html

2) DSpace has been tested on multiple UNIX platforms (including Linux, hp/ux, Solaris), as well as on MacOS and Windows.

3) Institutions using DSpace are experimenting with various database systems, including DB2, MySQL, and Oracle.

4) While DSpace ships with Apache and Tomcat, the system will work run with any web server and java servlet engine. It has also been tested with JBOSS and others.

5) Fifteen DSpace implementations are in full production worldwide, and over 115 additional implementations are in progress (worldwide).

6) Updating script requires some manual changes.

7) For each major module.

8) Uploads compressed files, but doesn't uncompress them.

9) METS in development.

10) Requires some programming.

11) Via Google or customized Lucene implementation.

12) Through the SourceForge system.

Eprints Notes

1) Designed to run in most UNIX environments.
2) Apache 2.0 compatibility in development.
3) Does not use Javascript. CSS support preferred, but not essential.
4) PERL programmer requirements depend on the extent of customization an institution requires.
5) 122 running v2; 18 running v1.1.
6) UK, Ireland, India, Italy, Brazil, Australia, USA, Canada, France, Austria, Sweden, Germany, Slovenia.
7) Updating script requires some manual changes to configuration files.
8) Can update system without overwriting modifications to page templates, emails, help pages, and search pages.
9) Can be modified to use other systems, e.g., LDAP.
10) State of files is stored in SQL database.
11) Default. Submission roles can be modified and/or extended.
12) Could be configured to provide this functionality.
13) Planned.
14) Default formats: PostScript, PDF, ASCII, and HTML.
15) Batch processing (to improve system performance) in experimental stage.
16) Requires some programming.
17) Uses third-party software tools.

18) Full-text searching is available in release 2.3.x. Collateral full-text search engines have also been integrated by several Eprints installations. For example, the Indian Institute of Science (IISc), in Bangalore, India (http://eprints.iisc.ernet.in/) has integrated the Greenstone Digital Library Open Source Software to provide full-text searching, and the Archive SIC (Archive Ouverte en Sciences de l'Information et de la Communication) has implemented the htdig search engine (see: http://archivesic.ccsd.cnrs.fr/ search.html).

19) Currently only provides stemming for plurals. Fuller stemming in development.
20) Not set as a default, but is configurable by system administrator based on institution-supplied metadata.
21) System administrator can select sort fields. Search results can be sorted by any standard field.
22) Eprints has a wiki at <wiki.eprints.org>.

Fedora Notes

1) Tested on Linux, Solaris, all recent Windows, and MacOSX (requires some work). Generally will work with any machine hosting a 1.4 JRE.

2) Uses JDBC for database interoperability. Alternate database support requires JDBC driver and a custom module (Java) to be written. Requirements for this module are documented.

3) For simple system metadata and Dublin Core queries; full-featured search (full-text, XML query, etc) would have to be added separately.

4) If server is run on Unix. Setup requires little OS-specific knowledge. Unix knowledge helpful for setting up init scripts, etc.

5) Twenty monitored installations; over 3,000 software downloads.

6) 35 countries; 5 continents.

7) Two major APIs (Access & Management). Mixture of SOAP over HTTP and straight HTTP interfaces.

8) Only two roles: Administrator and Anonymous.

9) Both APIs support IP-based authentication. API-M also uses HTTP Basic. Plan to support more by late 2004.

10) Planned for late 2004. Currently administrator can disable content for anonymous access.

11) Via a METS template.

12) In Fedora, this would be a "distribution license" dissemination of an object, or just a simple datastream stored along with each object.

13) Fedora generates system usage and performance logfiles. While the Fedora logfiles are in XML, and could be analyzed by a reporting tool, such a tool is not built into the system.

14) Planned.

15) Planned.

16) Although any form of descriptive metadata can be stored in a Fedora repository (including non-XML forms), Fedora's metadata search facility operates only with the XML Dublin Core record for each object.

17) Very basic browse functionality is supported by each object's primary Dublin Core metadata and the search API.

18) An automatically-generated page of hyperlinks to "to-be-searchable" disseminations could be constructed using the search API.

19) Fedora's persistent, globally unique identifiers use URN-like syntax. They can be automatically assigned or pre-assigned. Linkage to centralized resolver planned.

20) Metadata, content, and behaviors can all be versioned (and any version can be viewed at any time), but there is no "branching" of versions.

i-Tor Notes

1) Recommended for installation.

2) i-Tor allows institutions to extend certain aspects of the interface using Java (for example, to create custom views for search results).

3) Planned for September 2004.

4) i-Tor is designed to provide an institution with the tools to set up any required workflow, but does not design a workflow into the system itself.

5) Uses Analog third-party software.

6) i-Tor allows data to be harvested directly from a researcher's home page. Assuming that the individual researcher's home pages are adequately maintained, this would eliminate the need for faculty to periodically update the repository.

7) Configurable by system administrator based on institution-supplied metadata.

10) In development.

11) SourceForge

MyCoRe Notes

1) Planned.

2) System requirements depend on collection size, number of expected users, database platform, etc.

3) Open Source environment: JDBC compliant RDBMS (tested: MySQL, PostgreSQL); XML:DB compliant databases (Apache Xindice, eXist, Tamino); and commercial environment: IBM Content Manager with IBM DB2.

4) Tested: Tomcat and Websphere.

5) XSL skills required for customizing user interface layout.

6) Ten installations for MILESS, the predecessor on which MyCoRe is based. Five unofficial MyCoRe test sites.

7) Possible via CVS.

8) Configurable.

9) Configurable. MyCoRe does not have a hard-coded metadata model. The system provides a Qualified Dublin Core data model as an example, but users can define/configure their own data models as required.

10) MyCoRe does not implement CNRI Handles, but has implemented a similar system, the German National Bibliuography Number. See: <http://www.persistent-identifier.de/aktivitaeten/URN-Implementation.php> (in German).

OPUS Notes

1) Some additions including paragraphs on Termination, Severability, and Integration.

2) Configurable database interface.

3) Search engine module can be substituted with minor effort. Has been tested with Google site search.

4) PHP programmer needed if institution intends to add new features to the system.

5) Requires some manual changes to configuration files.

6) Generic passwords for athors outside campus IP ranges.

7) Planned for release 3.0.

8) Requires third-party software like Analog.

9) Requires script modifications.

10) Except full text index.

11) Browsing by document type and by faculty/institute possible.

12) Only in full text search.

13) Uses URN as persistent identifiers.

14) Will be developed in cooperatioon with the German National Library.