

Open Society Institute



**A Guide to
Institutional Repository Software**

1st Edition

October 2003

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: Erik Groeneveld of Seek You Too B.V. (i-Tor); Christopher Gutteridge of the University of Southampton (Eprints.org); Henk Harmsen 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 of Tilburg University (ARNO); and MacKenzie Smith and Richard Rogers of MIT (DSpace).

Additionally, Lilian van der Vaart of DARE (a SURF program), Martin Feijen of Innervation (consultant to DARE), Susan Gibbons (University of Rochester), and William Nixon (University of Glasgow) generously provided valuable feedback and insight.

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

This work is licensed under the Creative Commons License Attribution-NoDerivs 1.0 (<http://creativecommons.org/licenses/by-nd/1.0>). OSI permits others to copy, distribute, display, and perform the work. In return, licensees must give the original author credit. In addition, OSI permits others to copy, distribute, display and perform only unaltered copies of the work — not derivative works based on it.

© 2003, Open Society Institute, 400 West 59th Street, New York, NY 10019

Prepared by Raym Crow
Chain Bridge Group

703.536.7447 ▪ crow@chainbridgegroup.com ▪ www.chainbridgegroup.com

A Guide to Institutional Repository Software

CONTENTS

1.0 Introduction

1.1 Document Purpose

1.2 Document Scope

2.0 System Descriptions

2.1 Summary System Descriptions

2.2 Feature & Functionality Table

1) INTRODUCTION

1.1 Document Purpose

Universities and research centers throughout the world are actively planning the implementation of institutional repositories. Such planning 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 document to help organizations with one facet of their repository planning: selecting a software system that best satisfies an institution's needs. An institution's system needs will be driven by its content policies and by the procedures required to implement those policies. A well designed and carefully planned repository program can function well with any of the systems discussed here—but none of these systems can help a poorly designed or inadequately planned repository succeed. Therefore, where necessary, the notes to the System Feature & Functionality Table (Section 2.2) attempt to explain the relevance of system technical features in the context of a repository's broader planning, design, and policy framework.

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. Systems that require proprietary software components—such as a commercial database management system—have not been included.¹
- They comply with 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 discussed here—CDSware, DSpace, Eprints, i-Tor, and MyCoRe—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 discusses 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 its particular requirements.

The System Description section has two parts: 1) a summary description of each system (Section 2.1) provides a brief overview, contact information, and links for further information. 2) a Feature & Functionality Table (Section 2.2) provides additional detail on specific system functionality. The systems described here were developed with varying design philosophies and goals. The inclusion of a feature in Section 2.2 does not indicate that the functionality is a *sine qua non* of 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. Further, these

¹ However, for some of the systems included here, use of proprietary software (for example, Oracle or DB2) and/or operating systems (for example, Windows, Solaris) is optional.

systems are evolving rapidly, and readers should refer to a system's web site or contact person for the latest information on its features and functionality.

2) SYSTEM DESCRIPTIONS

2.1 Summary System Descriptions

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 and supports electronic preprint servers, online library catalogs, and other web-based document depository systems. CERN uses CDSware to manage over 450 collections of data, comprising over 620,000 bibliographic records and 250,000 full-text documents, including preprints, journal articles, books, and photographs.

CDSware was built to handle very large repositories holding disparate types of materials, including multimedia content catalogs, museum object descriptions, confidential and public sets of documents, etc. 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

There are two CDSware-related mailing lists:

- project-cdsware-announce@cern.ch
Available from <<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 from <<http://cdsware.cern.ch/lists/project-cdsware-users/archive/>>
Unmoderated, potentially high-volume mailing list, intended for discussion among users and developers of CDSware.

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 1.1.1 of the software was released in August 2003. 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, and various of its adopters are actively engaged in research and development in this area, which will, over time, allow DSpace adopters to offer services both for housing and making accessible the research material of their institutions, but also to maintain its utility 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 from <<http://dspace.org/technology/architecture.pdf>>.
- 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 from <<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.

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,² the first version of the system was publicly released in late 2000. The project was originally

² Eprints was written by Rob Tansley (based on the CogPrints software, which was written by Matt Hemus), and subsequently upgraded and maintained by Chris Gutteridge.

sponsored by CogPrints, but is now supported by JISC as part of the Open Citation Project and by NSF.

Eprints worldwide installed based affords an extensive support network for new implementations. The size of the installed base for Eprints suggests that an institution can get the 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
cjpg@ecs.soton.ac.uk
+44 (0)23 8059 4833
<http://software.eprints.org/>

Additional Eprints.org Information

- Nixon, William J. "The evolution of an institutional e-prints archive at the University of Glasgow." *Ariadne* 32 (July 8, 2002).
Available at: <http://www.ariadne.ac.uk/issue32/eprint-archives/intro.html>

Article recounts the experience of the University of Glasgow in setting up an institutional repository using the eprints.org software.
- 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/>.

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).³ NIWI calls i-TOR “a web technology by which various types of information can be presented 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 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

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>

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, thus “CoRe”). The bundle is designed to be configurable and adaptable to local requirements (hence, the “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 repositories, OAI functionality, audio/video streaming support, file management, online metadata editors etc.

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

³ See: www.niwi.knaw.nl

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/engl/index.html>

2.2 Feature & Functionality Table

Feature	CDSware	DSpace	Eprints	i-Tor	MyCoRe
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
1.2 Z39.50 protocol compliant	No	No	No	No	No ¹
1.3 Open source license ¹	GNU GPL	BSD	GNU GPL	GNU GPL	GNU GPL
1.4 Latest version release date	Apr-02	Aug-03	Mar-02	Aug-03	Oct 03
1.5 Latest version number	0.0.9	1.1.1	2.2.1	1.1.4	1.0
2.0 Hardware					
2.1 Minimum hardware requirements ²	No specific requirements ¹	No specific requirements ¹	No specific requirements	No specific requirements	No specific requirements ²
2.2 SAN support ³		Yes	Yes		
3.0 Software					
3.1 Operating system (tested)	Linux/Solaris	UNIX/MacOS/Windows ²	GNU/Linux/Solaris ¹	Linux/Windows	AIX/Windows/Linux/Solaris
3.2 Programming language	Python/PHP	Java	Perl	Java	Java
3.3 Database	MySQL	PostgreSQL ³	MySQL	MySQL & Oracle	MySQL, PostgreSQL; XML:DB compliant; Commercial databases ³
3.4 Web server	Apache/PHP, Python	Any ⁴	Apache 1.3 ²	Jetty	Apache
3.5 Java servlet engine		Any ⁴	N/A	Jetty	Any ⁴
3.6 Search engine	cdsware ²	Lucene	N/A	Lucene	Via JDBC and XML:DB
3.7 Other	WML: Website META Language	OAI:Cat	N/A		Apache Ant build tool
4.0 Clients supported	All HTML 4.0 clients	All web browsers	Netscape, Mozilla, IE, Lynx ³	All HTML 4.0 clients	All web browsers
5.0 Staff requirements⁴					
5.1 UNIX systems administrator	Yes	Yes	Yes	Recommended ¹	Recommended
5.2 Java programmer	No	Recommended	No	No	Recommended ⁵
5.3 PERL programmer	No	No	Recommended ⁴	No	No
5.4 Python programmer	No ³	No	No	No	No
6.0 Installed base					
6.1 Number of installations	7+ ⁴	10+ ⁵	106 ⁵	10	10 ⁶
6.2 Geographic coverage	Europe & US ⁵	Worldwide	Worldwide ⁶	Netherlands	Germany & Sweden

Feature	CDSware	DSpace	Eprints	i-Tor	MyCoRe
Repository & System Administration					
7.0 Set-up/Installation					
7.1 Automated installation script	Yes	Yes	Yes	Yes	Yes
7.2 System update script	Yes	Yes ⁶	Yes ⁷	No	Via CVS repository
7.3 Update system update without overwriting customized features ⁵	Yes		Yes ⁸	Yes	Yes ⁷
8.0 Module-level API(s)⁶	Yes ⁶	Yes ⁷	Yes	Yes ²	Yes
9.0 User registration, authentication & password administration					
9.1 Password administration	Yes	Yes	Yes	Yes	
9.1.1 System-assigned passwords	Yes ⁷	Yes	No	No	
9.1.2 User selected passwords	Yes	Yes	Yes	Yes	
9.1.3 Forgotten password function ⁷	Yes	No	Yes	No	
9.2 User registration verification/Other security mechanisms ⁸	MySQL table/Apache ACL	email/X.509	MySQL table ⁹	No	RDBMS table
9.2.1 Edit user profile	Yes	No	Yes	Yes	
9.3 Limit Access by User Type ⁹	Yes	Yes	Yes	No ³	
9.4 Multiple Authentication Methods ¹⁰	Yes	Yes	No	No ⁴	
9.5 Limit Access at File/Object Level ¹¹	Yes	Yes	Yes	Yes	No
10.0 Content Submission Administration					
10.1 Define multiple collections within same instance of system ¹²	Yes ⁸	Yes	Yes	Yes	
10.1.1 Set different submission parameters for each collection ¹³	Yes				
10.1.2 Home page for each collection	Yes ⁹	Yes	No	No	
10.2 Submission Stages¹⁴	Submit, Modify, Revise, Approve, etc. ¹⁰	Assemble, Pending, Approved		Yes ⁵	No ¹
10.2.1 Segregated submission workspace ¹⁵	Yes	Yes	Yes ¹⁰	Yes ⁵	
10.2.2 Submission roles ¹⁶	Submitters, Moderators, Reviewers, Approvers, Administrators	Submitters, Reviewers, Approvers, Editors	User, Editor, Administrator ¹¹	Yes ⁵	
10.2.3 Configurable submission roles within collections ¹⁷	Yes	Yes		Yes ⁵	
10.3 Submission Support					
10.3.1 Email notification for submitters ¹⁸	Yes ⁹	Yes	Yes	Yes	No
10.3.2 Email notification for content administrators ¹⁹	Yes ⁹	Yes	Yes	Yes	No
10.3.3 Personalized system access for registered users ²⁰	Yes	Yes	Yes	Yes	No
10.3.3.1 View pending content submissions ²¹	Yes	Yes	Yes	n/a	No
10.3.3.2 View approved content ²²	Yes	Yes	Yes	n/a	No
10.3.3.3 View pending content administration tasks ²³	Yes	Yes		n/a	No
10.3.4 Distribution license ²⁴					
10.3.4.1 Request distribution license ²⁵	No	Yes	No	No	
10.3.4.2 Store distribution license with content ²⁶	No	Yes	No ¹²	No	
11.0 System generated usage statistics and reports					
11.1 System-generated usage statistics ²⁷	No ¹¹	Yes	No ¹³	Yes ⁶	No
11.2 Usage reports ²⁸	No	Yes	No	Yes	No

Feature	CDSware	DSpace	Eprints	i-Tor	MyCoRe
Content Management					
12.0 Content Import/Export					
12.1 Upload compressed files	Yes	Yes ⁸	Yes	Yes	No ¹
12.2 Upload from existing URL	Yes	No	Yes	Yes ⁷	No ¹
12.3 Volume import for objects ²⁹	Yes	Yes	Yes	No	Yes
12.4 Volume import for metadata ³⁰	Yes	Yes	Yes	Yes	Yes
12.5 Volume export/content portability ³¹	Yes	Yes	Yes	No ⁸	Yes
13.0 Document/Object Formats					
13.1 Approved file format function ³²	Yes	Yes	Yes	No	No
13.2 File formats ingested ³³	All ¹²	All	All ¹⁴	All	All
13.3 Submitted items can comprise multiple files ³⁴	Yes	Yes	Yes		Yes
14.0 Metadata					
14.1 Basic metadata schema ³⁵	Standard Marc21	Qualified Dublin Core	Dublin Core	Any	Qualified Dublin Core ⁸
14.2 Support for extended metadata ³⁶	Yes	Custom	Yes	Any	Any ⁹
14.3 Metadata review support ³⁷	Yes	Yes	Accept, Edit, Bounce (require changes), Delete	No	No
14.4 Metadata export ³⁸	OAI-Marc export	Custom XML schema ⁹	Custom XML Schema	Yes	Yes
14.5 Allow metadata harvesting ³⁹	Yes	Yes	Yes	Yes ³	Yes
14.6 Add/delete metadata fields	Yes			Yes ³	Yes
14.7 Set default values for metadata ⁴⁰	Yes			Yes ³	
14.8 Supports Unicode character set for metadata	Yes	Yes	Yes	No	Yes
15.0 Real-time updating and indexing of accepted content	Yes	Yes	Yes ¹⁵	Yes	Yes

Feature	CDSware	DSpace	Eprints	i-Tor	MyCoRe
Dissemination (User Interface & Search Functionality)					
17.0 User Interface					
17.1 Modify interface "look & feel" ⁴¹	Yes	Yes ¹⁰	Yes ¹⁶	Yes	Yes
17.2 Apply a custom header/footer to static or dynamic pages	Yes ¹³	No	Yes	Yes	Yes
17.3 Supports multiple language interfaces	Yes	Yes	Yes	Yes	Yes
17.4 End user document folders ⁴²	Yes	No	No	Yes	
17.5 Discussion forum support ⁴³	No ¹⁴	No	Yes ¹⁷	Yes	No
18.0 Search Capability					
18.1 Full text ⁴⁴	Yes	Yes ¹¹	No ¹⁸	Yes	No ¹⁰
18.1.1 Boolean logic	Yes	No	No	Yes	
18.1.2 Truncation/wildcards ⁴⁵	Yes	No	No	Yes	
18.1.3 Word stemming ⁴⁶	No	No	No ¹⁹	No	
18.2 Search all metadata ⁴⁷	Yes	Yes	Yes	Yes	Yes
18.2.1 Boolean logic	Yes			Yes	Yes
18.2.2 Truncation/wildcards	Yes	Yes		Yes	
18.2.3 Word stemming	No	Yes		Yes	Yes
18.3 Search selected metadata fields ⁴⁸	Yes	Yes	Yes	Yes	Yes
18.4 Browse					
18.4.1 By author	Yes	Yes	Yes ²⁰	Yes ⁹	Yes
18.4.2 By title	Yes	Yes	Yes ²⁰	Yes ⁹	Yes
18.4.3 By issue date	Yes	Yes	Yes ²⁰	Yes ⁹	Yes
18.4.4 By subject term	Yes	No	Yes ²⁰	Yes ⁹	Yes
18.4.5 By collection	Yes	Yes	Yes ²⁰	Yes ⁹	Yes
18.5 Sort search results					
18.5.1 By author	Yes	No	Yes	Yes	Yes
18.5.2 By title	Yes	No	Yes	Yes	Yes
18.5.3 By issue date	Yes	No	Yes	Yes	Yes
18.5.4 By relevance	No	No	No	Yes	
18.5.5 By other	Any metadata field	No	Yes ²¹	Yes ⁹	Yes
19.0 Indexed by Google/Other Search Engines ⁴⁹	Possible ¹⁵	Yes		Yes	Possible
Archiving					
20.0 Persistent document identification⁵⁰					
20.1 System-assigned identifiers	Yes	Yes	Yes	Yes	Yes
20.2 CNRI Handles ⁵¹		Yes	Yes	No	Yes
21.0 Data preservation support					
21.1 Defined digital preservation strategy ⁵²	Yes ¹⁶	Yes	No	No	No ¹
21.2 Preservation metadata support (see also 14.2) ⁵³	Yes ¹⁷	Yes	No	No	No ¹
21.3 Data integrity checks	No	MD5 checksum	MD5 checksum	No	MD5 checksum
22.0 Object history/Version control	Versioning system	ABC Harmony data model	Some	No	No ¹
System Maintenance					
23.0 System support					
23.1 Documentation/manual	Yes	Yes	Yes	Yes ³	Yes
23.2 Listserv	Yes	Yes	Yes	Yes ³	Yes
23.3 Bug track/feature request system	Yes	Yes ¹²	No	Yes ³	No
23.4 Formal support/help desk	For fee	No	No	No	No

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. DSpace intends to add any third-party components under the same terms.
- 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 allow 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. All the systems 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 simplify 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.
- 423) 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 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

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
- 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) Ten DSpace implementations are in full production worldwide, and over 100 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) 88 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 under development. While Eprints.org does not yet have an integrated full-text search capability, collateral full-text search engines have 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.

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 December 2003.
- 4) Does not support validation by IP.
- 5) 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.
- 6) Uses Analog third-party software.
- 7) 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.
- 8) Planned.
- 9) Configurable by system administrator based on institution-supplied metadata.

10) In development.

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) Planned, via Lucene. Some limited text search functionality is given by the underlying XML:DB API MyCoRe uses (for example for searching in the abstract/description of objects).