

2.2 Feature & Functionality Table

Feature	ARNO	CDSware	DSPACE	Eprints	Fedora	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	OAI-PMH 2.0	OAI-PMH 2.0
1.2 Z39.50 protocol compliant	No	No	No	No	No	No	No ¹
1.3 Open source license ¹	TBD	GNU GPL	BSD	GNU GPL	MPL	GNU GPL	GNU GPL
1.4 Latest version release date	Dec-03	Apr-02	Aug-03	Mar-02	Dec-03	Aug-03	Oct 03
1.5 Latest version number	1.0	0.0.9	1.1.1	2.2.1	1.2	1.1.4	1.0
2.0 Hardware							
2.1 Minimum hardware requirements ²	No specific requirements	No specific requirements ¹	No specific requirements ¹	No specific requirements	No specific requirements	No specific requirements	No specific requirements ²
2.2 SAN support ³			Yes	Yes	Yes		
3.0 Software							
3.1 Operating system (tested)	Linux/Solaris	Linux/Solaris	UNIX/MacOSX/ Windows ²	GNU/Linux/Solaris ¹	Unix/MacOSX/Windows ¹	Linux/Windows	AIX/Windows/Linux/ Solaris
3.2 Programming language	Perl	Python/PHP	Java	Perl	Java	Java	Java
3.3 Database	Oracle 8i ¹	MySQL	PostgreSQL ³	MySQL	MySQL/McKoi/Oracle ²	MySQL & Oracle	MySQL, PostgreSQL; XML:DB compliant; Commercial databases ³
3.4 Web server	Apache	Apache/PHP, Python	Any ⁴	Apache 1.3 ²	Tomcat 4.1	Jetty	Apache
3.5 Java servlet engine			Any ⁴	N/A	Tomcat 4.1	Jetty	Any ⁴
3.6 Search engine	Unix & SQL command-line	cdsware ²	Lucene	N/A	Database ³	Lucene	Via JDBC and XML:DB
3.7 Other		WML: Website META Language	OAIcat	N/A			Apache Ant build tool
4.0 Clients supported	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	All HTML 4.0 clients	All web browsers
5.0 Staff requirements⁴							
5.1 UNIX systems administrator	Yes	Yes	Yes	Yes	For setup ⁴	Recommended ¹	Recommended
5.2 Java programmer	No	No	Recommended	No	Recommended	No	Recommended ⁵
5.3 PERL programmer	Recommended	No	No	Recommended ⁴	No	No	No
5.4 Python programmer	No	No ³	No	No	No	No	No
6.0 Installed base							
6.1 Number of installations	7	7 ⁴	15 ⁵	106 ⁵	20 ⁵	10	10 ⁶
6.2 Geographic coverage	Netherlands	Europe & US ⁵	Worldwide	Worldwide ⁶	Worldwide ⁶	Netherlands	Germany & Sweden

Feature	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe
Repository & System Administration							
7.0 Set-up/Installation							
7.1 Automated installation script	Yes	Yes	Yes	Yes	Yes	Yes	Yes
7.2 System update script	Yes	Yes	Yes ⁶	Yes ⁷	Yes	No	Via CVS repository
7.3 Update system update without overwriting customized features ⁵	Yes ²	Yes	Yes	Yes ⁸	Yes	Yes	Yes ⁷
8.0 Module-level API(s)⁶	No	Yes ⁶	Yes ⁷	Yes	Yes ⁷	Yes ²	Yes
9.0 User registration, authentication & password administration							
9.1 Password administration	Yes	Yes	Yes	Yes	Yes	Yes	
9.1.1 System-assigned passwords	No	Yes ⁷	Yes	No	No	No	
9.1.2 User selected passwords	Yes	Yes	Yes	Yes	Yes	Yes	
9.1.3 Forgotten password function ⁷	Yes ³	Yes	Yes	Yes	No	No	
9.2 User registration verification/Other security mechanisms ⁸	LDAP and/or ARNO registry	MySQL table/Apache ACL	email/X.509	MySQL table ⁹	No	No	RDBMS table
9.2.1 Edit user profile	No	Yes	Yes	Yes	No	Yes	
9.3 Limit Access by User Type ⁹	Yes	Yes	Yes	Yes	Yes ⁸	No ³	
9.4 Multiple Authentication Methods ¹⁰	Yes	Yes	Yes	No	Yes ⁹	No ⁴	
9.5 Limit Access at File/Object Level ¹¹	Yes	Yes	Yes	Yes	No ¹⁰	Yes	No
10.0 Content Submission Administration							
10.1 Define multiple collections within same instance of system ¹²	Yes	Yes ⁸	Yes	Yes	Yes	Yes	
10.1.1 Set different submission parameters for each collection ¹³	Yes	Yes			Yes ¹¹		
10.1.2 Home page for each collection	No	Yes ⁹	Yes	No	Yes	No	
10.2 Submission Stages¹⁴		Submit, Modify, Revise, Approve, etc. ¹⁰	Assemble, Pending, Approved		Ingest, Create, Modify, Activate, Deactivate	Yes ⁵	No ¹
10.2.1 Segregated submission workspace ¹⁵	Yes	Yes	Yes	Yes ¹⁰	Yes	Yes ⁵	
10.2.2 Submission roles ¹⁶	Contributors, Editors, Administrators, Site Managers	Submitters, Moderators, Reviewers, Approvers, Administrators	Submitters, Reviewers, Approvers, Editors	User, Editor, Administrator ¹¹	Administrator	Yes ⁵	
10.2.3 Configurable submission roles within collections ¹⁷	Yes	Yes	Yes		No	Yes ⁵	
10.3 Submission Support							
10.3.1 Email notification for submitters ¹⁸	Only during registration	Yes ⁹	Yes	Yes	No	Yes	No
10.3.2 Email notification for content administrators ¹⁹	Yes	Yes ⁹	Yes	Yes	No	Yes	No

10.3.3 Personalized system access for registered users ²⁰	Yes	Yes	Yes	Yes	No	Yes	No
10.3.3.1 View pending content submissions ²¹	Yes	Yes	Yes	Yes	No	n/a	No
10.3.3.2 View approved content ²²	Yes	Yes	Yes	Yes	No	n/a	No
10.3.3.3 View pending content administration tasks ²³	Yes	Yes	Yes		No	n/a	No
10.3.4 Distribution license ²⁴							
10.3.4.1 Request distribution license ²⁵	No	No	Yes	No	Yes ¹²	No	
10.3.4.2 Store distribution license with content ²⁶	No	No	Yes	No ¹²	Yes	No	
11.0 System generated usage statistics and reports							
11.1 System-generated usage statistics ²⁷	Yes	No ¹¹	Yes	No ¹³	Yes ¹³	Yes ⁶	No
11.2 Usage reports ²⁸	No	No	Yes	No	No ¹⁴	Yes	No

Feature	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe
Content Management							
12.0 Content Import/Export							
12.1 Upload compressed files	Yes	Yes	Yes ⁸	Yes	Yes	Yes	No ¹
12.2 Upload from existing URL	Yes	Yes	No	Yes	Yes	Yes ⁷	No ¹
12.3 Volume import for objects ²⁹	Yes	Yes	Yes	Yes	Yes	No	Yes
12.4 Volume import for metadata ³⁰	Yes	Yes	Yes	Yes	Yes	Yes	Yes
12.5 Volume export/content portability ³¹	Yes	Yes	Yes	Yes	Yes	No ⁸	Yes
13.0 Document/Object Formats							
13.1 Approved file format function ³²	Yes	Yes	Yes	Yes	No ¹⁵	No	No
13.2 File formats ingested ³³	All	All ¹²	All	All ¹⁴	All	All	All
13.3 Submitted items can comprise multiple files ³⁴	Yes	Yes	Yes	Yes	Yes		Yes
14.0 Metadata							
14.1 Basic metadata schema ³⁵	Dublin Core	Standard Marc21	Qualified Dublin Core	Dublin Core	Dublin Core	Any	Qualified Dublin Core ⁸
14.2 Support for extended metadata ³⁶	Yes	Yes	Custom	Yes	Yes	Any	Any ⁹
14.3 Metadata review support ³⁷	Yes	Yes	Yes	Accept, Edit, Bounce (require changes), Delete	No	No	No
14.4 Metadata export ³⁸	Yes	OAI-Marc export	Custom XML schema ⁹	Custom XML Schema	Yes	Yes	Yes
14.5 Disallow metadata harvesting ³⁹	Yes	Yes	Yes	Yes	Yes	Yes	Yes
14.6 Add/delete metadata fields	Yes	Yes	Yes	Yes	Yes	Yes ³	Yes
14.7 Set default values for metadata ⁴⁰	Yes	Yes	Yes		No	Yes ³	
14.8 Supports Unicode character set for metadata	Partial ⁴	Yes	Yes	Yes	Yes	No	Yes
15.0 Real-time updating and indexing of accepted content	No	Yes	Yes	Yes ¹⁵	Yes	Yes	Yes

Feature	ARNO	CDSware	DSpace	Eprints	Fedora	i-Tor	MyCoRe
Dissemination (User Interface & Search Functionality)							
16.0 User Interface							
16.1 Modify interface "look & feel" ⁴¹	No	Yes	Yes ¹⁰	Yes ¹⁶	Yes	Yes	Yes
16.2 Apply a custom header/footer to static or dynamic pages	No	Yes ¹³	No	Yes	Yes	Yes	Yes
16.3 Supports multiple language interfaces	No	Yes	Yes	Yes	Yes	Yes	Yes
16.4 End user document folders ⁴²	No	Yes	No	No	No	Yes	
16.5 Discussion forum support ⁴³	No	No ¹⁴	No	Yes ¹⁷	No	Yes	No
17.0 Search Capability							
17.1 Full text ⁴⁴	No	Yes	Yes ¹¹	No ¹⁸	No	Yes	No ¹⁰
17.1.1 Boolean logic	No	Yes	No	No	No	Yes	
17.1.2 Truncation/wildcards ⁴⁵	No	Yes	No	No	No	Yes	
17.1.3 Word stemming ⁴⁶	No	No	No	No ¹⁹	No	No	
17.2 Search all descriptive metadata ⁴⁷	No	Yes	Yes	Yes	Yes ¹⁶	Yes	Yes
17.2.1 Boolean logic	No	Yes	Yes		No	Yes	Yes
17.2.2 Truncation/wildcards	No	Yes	Yes		Yes	Yes	
17.2.3 Word stemming	No	No	Yes		No	Yes	Yes
17.3 Search selected metadata fields ⁴⁸	No	Yes	Yes	Yes	Yes	Yes	Yes
17.4 Browse							
17.4.1 By author	No	Yes	Yes	Yes ²⁰	Yes ¹⁷	Yes ⁹	Yes
17.4.2 By title	No	Yes	Yes	Yes ²⁰	Yes	Yes ⁹	Yes
17.4.3 By issue date	No	Yes	Yes	Yes ²⁰	Yes	Yes ⁹	Yes
17.4.4 By subject term	No	Yes	No	Yes ²⁰	Yes	Yes ⁹	Yes
17.4.5 By collection	No	Yes	Yes	Yes ²⁰	Yes	Yes ⁹	Yes
17.5 Sort search results							
17.5.1 By author	No	Yes	No	Yes	No	Yes	Yes
17.5.2 By title	No	Yes	No	Yes	No	Yes	Yes
17.5.3 By issue date	No	Yes	No	Yes	No	Yes	Yes
17.5.4 By relevance	No	No	No	No	No	Yes	
17.5.5 By other	No	Any metadata field	No	Yes ²¹	No	Yes ⁹	Yes
18.0 Indexed by Google/Other Search Engines⁴⁹	No	Possible ¹⁵	Yes		Possible ¹⁸	Yes	Possible
Archiving							
19.0 Persistent document identification⁵⁰							
19.1 System-assigned identifiers	Yes	Yes	Yes	Yes	Yes ¹⁹	Yes	Yes
19.2 CNRI Handles ⁵¹	No		Yes	Yes	No	No	Yes
20.0 Data preservation support							
20.1 Defined digital preservation strategy ⁵²	No ⁵	Yes ¹⁶	Yes	No	Yes	No	No ¹

20.2 Preservation metadata support (see also 14.2) ³³	Yes	Yes ¹⁷	Yes	No	Yes	No	No ¹
20.3 Data integrity checks	No	No	MD5 checksum	MD5 checksum	SIP schema validation	No	MD5 checksum
21.0 Object history/Version control	Versioning system for both metadata & objects	Versioning system	ABC Harmony data model	Some	Linear ²⁰	No	No ¹

System Maintenance

22.0 System support

22.1 Documentation/manual	Yes ⁶	Yes	Yes	Yes	Yes	Yes ³	Yes
22.2 Listserv	Yes	Yes	Yes	Yes	Yes	Yes ³	Yes
22.3 Bug track/feature request system	No	Yes	Yes ¹²	No	Yes	Yes ³	No
22.4 Formal support/help desk	No	For fee	No	No	Yes	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 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 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

ARNO Notes

- 1) Port planned to PostgreSQL or other Open Source DBMS.
- 2) Excluding changes in source code.
- 3) For users registered via LDAP.
- 4) Full support in development.
- 5) Under development in conjunction with DARE project.
- 6) 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) 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.

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 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).