Digital Library Federation
Digital Library Content and Course Management Systems:
Issues of Interoperation
July, 2004

Digital Repository Summary Checklist of Service Requirements, with Recommended Best Practices

Kerry Blinco, IMS Australia

NOTE: This document summarizes in a more succinct and easy to use form the recommendations of the Work Group on repository service requirements. For the overall rationale for the checklist and fuller discussion of the checklist items, see Appendix 2 of this report.

Scholarship and higher education increasingly depend on digital information, and the online sources that provide them, for research and teaching. These sources vary greatly in size, focus, function, and scope. Valuable teaching and research materials might be found in a dataset collection on a departmental web site, in a repository of images run by a university library, or in a licensed commercial database of journal articles. Large numbers of these data sources, often known as **digital repositories** now exist.

To make the most effective use of digital content in teaching, learning applications need to be able to easily interoperate with multiple digital repositories so that teachers and students can discover, access, view, quote, adapt, and evaluate appropriate learning material. Unfortunately, many data sources have not been designed to interoperate with other repositories or with learning applications, and are instead designed primarily as isolated "content silos" that can only be used through a single repository-specific interface. Information in such sources is therefore difficult to gather together and adapt effectively for research and teaching. Greater repository interoperability will not only help students and teachers, but will also increase the value of repositories that are interoperable with learning applications, since users will gravitate towards systems that make it easy to gather necessary information for research or teaching.

An awareness of the need for interoperation of repositories of quality content with systems supporting learning and teaching has been growing over the past few years. In order to further progress in this area, the Andrew W. Mellon Foundation provided support for an *ad hoc* group of digital librarians, course management system developers, and publishers to meet and discuss useful next steps to increase the integration of existing digital resources into the working environments of instructors in higher education. The Group co-chaired by Dale Flecker of Harvard University and Neil McLean of IMS Australia, produced a report summarizing the work of the Group as of March, 2004 [http://www.diglib.org/pubs/cmsdl0407/].

The report includes the outputs of a working group formed to analyze in detail what services and practices repository owners should consider when designing their offerings. Based on these outputs, this Checklist includes the working group's recommendations together with a summary of the associated contextual discussion (for a more detailed discussion the full report of the working group is available in Appendix 2: http://www.diglib.org/pubs/cmsdl0407/cmsdl0407app2.htm).

This Checklist recommends:

- a set of essential services and features that any digital repository seriously intended for academic audiences must provide;

- other services and features that are desirable for interoperation with teaching and research applications; and

- current best practices and standards.

Intended primarily for those developing repository systems and those developing software that uses digital content to support teaching and learning, the Checklist should assist in understanding the features and services they should provide to be most useful to an academic and scholarly audience. Many of the requirements and recommendations given will benefit not only repositories hosted in the academic environment, but also repositories in the not-for-profit and the commercial sectors, and will apply to other uses of digital repositories as well. The working group identified two general areas of design important for interoperation and eleven services relevant to the discovery and reuse of digital resources. Five of these services were considered by the working group to be essential.

The overall thrust of the Checklist is that repositories and related information systems should:

- make themselves known to operators of learning applications in expected ways;

- follow standards and best practices in terms of access, search, metadata practices, and download support; and

- document their systems and policies so that others can configure their systems appropriately to interoperate with them.

Taken together, these steps should significantly ease the task of integrating information systems into the learning environment.

The Checklist identifies interoperability principles or features that are Essential or Desirable. Each principle or feature is further broken down into Required or Optional sub-parts.

THE CHECKLIST

A. General Design Principles

General design principles that repository services should follow in order to be accessible in useful ways from learning applications:

	Design Principle	**Context**	**Technical Recommendations**
1.	**Ensure the repository is as broadly available and widely accessible as possible** **(Desirable)**		
1.1	Provide standards based interfaces to the repository (Optional)	When a repository is exposing functions or data or other repositories or applications, standards based interfaces should be used.	See specific interface functions for technical recommendations.
1.2	Support accessibility standards and best practice (Optional or Required if mandated by Law)	Many jurisdictions have laws or policies that require accessible design for user interfaces. (e.g. Section 504 of the US Rehabilitation Act), see: W3C Policies Relating to Web Accessibility http://www.w3.org/WAI/Policy/)	The IMS Accessibility Special Interest Group has produced a number of documents and specifications promoting accessibility in learning: http://www.imsglobal.org/accessibility/index.cfm#version1
	⇒ Provide textual navigation capability	If the primary navigation method is non – textual, such as an image map, provide alternative or supplemental textual means of navigation.	
	⇒ Describe other accessibility features provided (Repository to describe)		The Trace Center has comprehensive guidelines on developing accessible software: http://trace.wisc.edu/world/computer_access/software/
	⇒ Describe any limitations for access by disabled users (Repository to describe)		
1.3	Use standard character sets	Character encodings that	

	and encodings (Optional)	are compatible with a wide range of languages are recommended.	
	⇒ Unicode support - encoding - character blocks (Repository to describe)		Unicode with UTF-8 encoding is recommended. Characters that conform to Unicode are also conformant to ISO/IEC 10646. Unicode is grouped into Code Blocks of related characters. Applications should provide support for those character code blocks required for the languages supported by the repository.
	⇒ Other character coding support (Repository to describe)	Describe other character sets and encodings supported.	E.g. ISO2022 Character code structure and extension techniques - Describe sets supported and initial settings
2.0	**Provide access controls that allow learners and learning applications to access functions and content** **(Desirable)**	While some digital repositories may expose all of their content for the world to see, many repositories, particularly those of commercial publishers, may only provide access to the content to trusted users or paid subscribers. They may also limit what users can do with their content.	
2.1	Make all repository functionality and content available for public (non-authenticated) access. (Optional)		
2.2	When access control is required, apply best practice principles (Optional)		
	⇒ Provide as much of the repository functionality and content as possible for non-authenticated access.		
	⇒ Authenticate only at point	If a user must be	

	of need.	authenticated to access functions or content, authentication should be required only at the point of need.	
	⇒ Document access and usage rights policies for collections and items in associated administrative metadata	Including rights policies in metadata makes it clear to users what they (and their colleagues and students) can do with items. Some repositories can use rights metadata to automatically manage access and usages. Some repositories may allow searches on rights criteria (by specifying, for example, that one is only interested in items that are free to access).	
	⇒ Support standard authentication and authorization technologies. (Repository to describe)	Standard mechanisms for authentication and authorization make it easier for learning applications to integrate with repository services and content.	Some examples include • Kerberos • LDAP • Proxy servers. • Public Key (X.509) certificates. • Virtual Private Networks (VPNs). • Institutional single sign-on services (e.g. WebISO, Pubcookie) • Shibboleth
	⇒ Integrate with institutional authentication and authorization systems. (Repository to describe)	Repositories should access institutional authentication systems to minimize the need for users to re-authenticate.	As Above

B. Repository Services and Features

For ease of discussion, the eleven recommended repository services and features have been grouped into four categories. These services and features directly enable searching, collecting, and importing and provide essential information that supports these and other activities. Metadata provides crucial information for searching, helps users identify and evaluate items for collection, and documents items when they are imported. Publicizing the policies and functions of a repository lets users understand the authority, reliability, and usability of the repository and its contents, which is crucial to understanding their usability in teaching and learning.

The services and features that the working group recommends a repository should provide can be grouped as follows (The services and features in **bold** are considered by the working group to be essential):

Discovering Content:

1. **Support search for items**.
2. **Provide standard or documented metadata for items**.
3. Support search via software agents.

Collecting Content:

4. **Provide stable references to items.**
5. Support citations (in recognized scholarly formats) for items.

Accessing Content:

6**. Provide ways to get and use item content**.
7. Provide views of item content.
8. Allow items to be copied into local systems.

Documentation:

9. **Document policies and functions of the repository.**
10. Make the repository, and its content, known to other applications.
11. Document the technical profile of the repository.

Not included in this list are features related to depositing items into a repository. Instead, the working group focused on the use of items **from** a repository by learning applications. After some consideration, features such as versioning support, usage statistics, or refinement of search results were omitted. While these can be useful features for repositories to support, they either have little to do with interoperation with learning applications, or were not seen as highly desired by content users at this time.

	Service / Feature	**Context**	**Technical Recommendations**
FINDING CONTENT			
1.	**Support Search for Items (Essential)**	The repository must provide an interface that allows users to locate the items that they need	
1.1	Basic querying and browsing (Required)		
	⇒ Locate items by items by	The repository must	

	their title and creator	provide query of descriptive metadata by title and creator as a minimum	
	⇒ Inventory all items in the repository	Users and software agents must be able to browse the full contents of a repository	
	- Inventories available to users		
	- Inventories available to programs that can index, list, or harvest the repository	See Services & Feature Section 3 for more details	
	- List all collections when there are component collections	If there are multiple collections, inventorying should list all collections	
	- Inventory component collections' contents as individual subsets	Browse those collections' contents as individual subsets.	
1.2	Advanced Query (Optional)		
	⇒ Perform general keyword queries		
	⇒ Query specific essential descriptive metadata fields		
	⇒ Query by		
	- Title		
	- Author/Creator		
	- Subject		
	- Date		
	- Any descriptive metadata for items		
	- Any administrative or technical metadata for items		
	- Format		
	⇒ Query based on content, e.g. full-text searching, and not just metadata		
1.3	Advanced Browsing: (Optional)		
	⇒ Browse by		
	- Title		
	- Author/Creator		

	- Subject		
	- Descriptive date		
	- Administrative date (date of ingest, date updated, etc.)		
	- Format (all images, texts, video, etc.)		
	⇒ Support Hierarchical browsing	e.g. collections with parent-child relationships and ordering.	
	⇒ Browse by other meaningful categories (Repository to describe)		
1.4	Find related items: (Optional)	Repositories should assist the users to find items related to ones that they find	
	⇒ Link to a query that contains items with similar metadata		
	⇒ Link to related items using "knowledge" such as usage history (Repository to describe)		
	⇒ Link to related information outside the repository via URL		
	⇒ Link to external content described by metadata via OpenURL		The OpenURL Framework for Context-Sensitive Services ANSI /NISO Z39.88
1.5	Alert Users when new material is available that matches their interests. (Optional)		Consider RSS (Rich Site Summary)
1.6	Present Search results to the User in a way that helps users select the material they want. (Optional)		
	⇒ Relevancy-based ranking		
	⇒ Sorted by title		
	⇒ Sorted by author		
	⇒ Other meaningful orders and displays (Repository to describe)		

2.	**Provide standard, documented metadata for items** **(Essential)**	Repositories must maintain the item-level metadata that describes its content. Such metadata helps users find appropriate content, and understand the nature of the content they find. Repositories must also present metadata in a form that end users can read and understand Repositories must also expose this metadata in standard machine-readable formats so that other applications can query, index, translate, and display it.	
2.1	Maintain item-level metadata that describes the items in the repository (Required)	Repositories need not contain standard records natively, but they should provide metadata in a format that can be mapped to standard metadata formats.	
2.2	Expose machine-readable metadata, processable by other applications (Required)	Dublin Core (DC) is the most ubiquitous standard that can be recommended as the minimal set of metadata elements that repositories should expose.	The Dublin Core Library Application Profile http://dublincore.org/documents/library-application-profile/ should be considered. At a minimum, unqualified Dublin Core encoded in the xml schema for Dublin Core in the OAI-PMH
2.2	Display User-comprehensible metadata (Repository to describe) (Required)		
	⇒ Minimum item level descriptive metadata – Title		
	⇒ minimum item-level technical metadata - MIME type	Technical metadata standards are generally format-specific since they are used primary for object life cycle management and long-term preservation.	
2.3	Repository does one or both of the following		

	(Required)		
	⇒ Export metadata in standard formats (Repository to describe)	Dublin Core as a minimum. For fuller descriptive metadata, a number of community-based standards should be considered.	• MARC (original or MARCXML) and MODS for bibliographic and general descriptive metadata • EAD for finding aids • TEI headers for text • The VRA Core for images • DDI for data sets. • The same standards, applied in conjunction with METS, should be considered for fuller administrative and technical metadata.
	⇒ Provide documentation for the conventions used for metadata (Repository to describe)	Documentation is particularly important where non-standard metadata, or metadata comprised of a composite of different schema is exposed. Both internal and exposed formats for metadata should be documented.	
2.4	Provide additional basic descriptive, technical, and administrative metadata. (Optional)	Administrative metadata should include basic information about the provenance and current stewardship of an item of content	
	⇒ Identifier		
	⇒ Author/creator		
	⇒ Date		
	⇒ Resource Type		
	⇒ Format		
	⇒ Rights		
2.5	Provide sufficient metadata to make it possible to cite an item in scholarly form (Optional)		
2.6	Provide structural metadata allowing complex items to be viewed and navigated in intelligible ways	Complex objects may include metadata that describes the structure and how it is navigated.	Standards for showing specific relationships between parts of an object are beginning to emerge in practice

	(Optional)		including METS from the library computing, IMS Content Packaging and the emerging ISO standard MPEG-21 for digital video and multimedia material.
2.7	Provide descriptive metadata to document the purpose, applicability, educational goals, and prerequisites of content (Optional)		The IEEE LOM, IMS Best Practice Guide for the LOM, SCORM and community application profiles of these specifications should be considered for full description of Learning Objects
	⇒ Available for searching		
	⇒ Available via browsing		
	⇒ Provide mechanism for the creation of additional metadata at ingest or creation time		
2.8	Provide rights information encoded in a rights expression language. (Optional)		The dominant rights expression languages in development are MPEG-21 REL, (based on XrML) and ODRL.
2.9	Ingest metadata in XML format associated with the metadata specifications supported. (Repository to describe) (Optional)	Repositories may of course store metadata differently internally for optimization	
3.	**Support Search via Software Agents** **(Desirable)**	Repositories should support search by software agents as well as users	
3.1	Provide standard search protocol interface to repository. (Optional)		Z39.50 is the most widely supported searching protocol in libraries today, and several meta-search products on the market support federated search via Z39.50. SRW, a more lightweight, XML-oriented search protocol based on Web Services and designed as a follow-up to Z39.50, is growing in popularity.

			SRW builds on Z39.50 semantics.
3.2	Support standard authentication mechanisms (if applicable) for software agent access to search services (Optional)		
3.3	Make Repository metadata harvestable (Optional)		
	⇒ via OAI-PMH		For harvesting, OAI-PMH is an important protocol. OAI-PMH requires repositories to provide metadata in unqualified Dublin Core, but it can also be used to expose any other XML-based metadata scheme, such as IMS Metadata or MODS.
	⇒ via web crawling		Public Internet search engines also harvest publicly readable repository items or metadata via ordinary HTTP, but such harvesting does not provide the structured metadata that can be exported using OAI.
	⇒ via other methods (Repository to describe)		For feeding portal systems directly, repository implementers may want to consider RSS, which also supports alerting.
COLLECTING CONTENT			
4.	**Provide stable references to items** **(Essential)**		
4.1	Provide a stable identifier for each item in the repository, usable by external systems to locate the item for as long it exists in the repository		

	(Required)		
4.2	Provide stable identifiers that are also unique (not used in other repositories) (Optional)		
4.3	Provide Persistent identifiers capable of outliving the repository (Repository to describe) (Optional)	Stable identifiers need to be supported in the repository itself. Debate continues over which specific approaches will prove dominant in the coming years, but choosing one of these approaches will help lessen the very real risk of broken links in the near term.	Underlying technology for persistent identifiers includes Handles, DOIs (Handles with additional constraints and support, including possible registration in systems like Crossref), and system-specific IDs. ARKs (Archival Resource Keys), persistent identifiers for archival objects. Whatever scheme is chosen for a repository, we recommend that stable IDs should be encoded in URLs for client resolution, since that is the only type of locator with wide native support now. PURL is a useful reference model for persistent URLs.
4.4	Identifiers point to:		We recommend that persistent IDs be set up to reference item records, so that users of content understand its nature and context. Repositories can also create stable (but not necessarily persistent) references pointing straight to content.
	⇒ Item records with metadata		
	⇒ Directly to content		

5.	**Support citations (in recognized scholarly formats) for items** **(Desirable)**	This capability helps users systematically collect and manage citations and bibliographic data for their own papers and publications.	
5.1	Support the creation or export of citations in recognized scholarly formats for items, based on their descriptive metadata. (Optional)	.	Multiple technical formats for citations may need to be provided.
	⇒ Via a text citation that can be easily copied and pasted		E.g. JSTOR uses a printer-friendly format - a simple text file with labels for all data fields (Title, Author, Stable URL, Abstract). This format contains no specially formatted text. This can be useful for cutting and pasting citation information.
	⇒ Via export to a saved citations list		For export to an eLearning system the IMS RLI specification should be considered. RLI is a web services specification for the interchange of resource lists and their association with programs of study.
	⇒ Directly to bibliographic software		Usually a tagged format. Commonly supported software includes EndNote, ProCite, Reference Manager, RefWorks
	⇒ Directly to spreadsheet software		E.g., a **tab-delimited** format can be used to import citations into a spreadsheet software such as Microsoft Excel
	⇒ Describe formats or software supported (Repository to describe)		
	⇒ Include persistent identifiers in metadata, if available		
	⇒ Thumbnail export available for cites of		

	image-based content		
ACCESSING CONTENT			
6.	**Provide ways to access and use content** **(Essential)**	Users need some means to get content that they have discovered through searching or browsing a repository so that they can use it in teaching and learning.	
6.1	Users with appropriate authorization able to : (Required)		
	⇒ Get the actual item content and then process it further		For repositories that interoperate with learning applications natively, a standard API (most probably SOAP based) for accessing items should be provided. An example is the Fedora Access API (API-A), which defines an interface for accessing digital objects stored in a repository. The Open Knowledge Initiative (OKI) is defining a Content Repository API to fulfill some of these functions. The IMS DRI specification includes Publish/store and request/deliver functions.
	⇒ Get views of that content that users can view, navigate, and analyze sufficiently to use in teaching and scholarship.	See Section B 7	
6.2	Selective access options provided for certain types of content (Optional)	If a repository supports full downloads, selective access may be possible simply through full retrieval, followed by some processing by the client in an additional application. But the application would have to understand how to then make the selection, and general standards for documenting selections are not mature at this point. To	

		support selective access at the repository level, the repository itself would need to understand different content formats.	
	⇒ Images provided with size, resolution, detail options (Repository to describe)	For example, images could be provided with different size and resolution, or with zooming and panning options. These functions could be handled with parameterized access requests ("show high-resolution TIFF version", "show a thumbnail", "show latest version"), and partial access ("show this data slice", "show this part of the image", "show streaming time stamp slice").	
	⇒ Recordings accessible in selected snippets	Large audio or video recordings might be usefully accessed in selected snippets.	
7.	**Provide views of item content** (Optional)	Not all digital content can be easily used simply by being copied or saved locally. Items containing large quantities of information, or those in unusual formats, may not be practical for teachers or students to import and work with directly. Additionally, copyright restrictions on some content may prevent its dissemination in full. In such cases, repositories may need to display content themselves. Views of various content types may or may not include full item export.	Different options can be offered based on criteria such as the item's MIME type or the presence or absence of multiple media files.
7.1	Content viewable via a web browser (Optional)	Repositories should provide a way for content to be viewed via a web browser.	Repositories should use MIME types to indicate the formats of the items they contain, so that they can be correctly viewed. Common MIME types should be supported by the repository's viewing

			interfaces, and correct MIME types should also be delivered to viewer applications.
	⇒ Directly		
	⇒ Translate content to HTML or other common browser format		
	⇒ Provide Plugin or applet (Repository to describe)		
	⇒ Display Metadata		
	- Include administrative metadata		
	- Date of creation or accession		
	- Collections item appears in		
	- Copyright information		
	- Other (Repository to describe)		
7.2	Repository supports navigation within complex items stored in the repository (Repository to describe) (Optional)		Repositories that ingest complex objects that include navigational metadata (e.g. from METS, CP and MPEG 21 Packages) support complex navigation on presentation to the user.
8	**Allow content to be copied into local systems (Highly Desirable)**		
8.1	Repository allows users to download content into their local applications (Optional)	Ideally, users should be able to get all metadata, along with all content bit-streams that are associated with the item. Repositories might suppress internal administrative or version data if that is not of interest to learning applications	To protect intellectual property or minimize the load on repositories some content may be downgraded to lesser resolution for export or limit the number or rate of downloads that are allowed. Packaging standards for learning objects and other repository items use many of the same standards that are used to record structural metadata: METS, IMS

			Content Packaging, and MPEG-21
	⇒ All metadata available to Users		
	⇒ All content bit-streams that are associated with the item available to users		
	⇒ Export downgrades (Repository to describe)		
	⇒ Any exceptions (Repository to describe)		
DOCUMENTATION			
9.	**Document policies and functions of the repository** **(Essential)**	It is essential for repository rights, restrictions, functions, and critical policies for security and privacy to be documented, at least informally or implicitly, at the repository level. These let users know what they can do with items they find in the repository. Human-readable documentation is especially important for repository-specific conventions.	An "Identify" call to the OAI Provider front-end on the repository supplies basic repository documentation. The minimum element set used to identify a provider may need to be extended to cover the categories of information desired here. Some such extended elements sets are found in the OAI Eprints schema http://www.openarchives.org/OAI/2.0/guidelines-eprints.htm and the RSLP Collection Description schema http://www.ukoln.ac.uk/metadata/rslp/schema/ Repositories intended to be trustworthy should consult RLG/OCLC's paper on trusted digital repositories http://www.rlg.org/pr/pr2002-repositories.html.
9.1	Critical policy documentation at the repository level (Required)		
	⇒ Copyrights and related rights		
	⇒ Security		
	⇒ Privacy		
9.2	Are these policies:		
	- formal		
	- informal but explicit		

	- implicit		
9.3	Access and usage rights documented at item metadata level (Optional)	Rights and restrictions are sometimes implicit in the access control. As an example, while most publisher sites are not providing detailed information on the rights for each item, they at least state somewhere that a subscription is required, and give terms of subscription and use to those who ask about it. Conventions must be documented so that users and applications understand how to interpret the metadata.	
9.4	Metadata conventions documented (Repository to describe) (Optional)	If the repository does not use standard metadata, it must document its metadata	In some cases, such as in Qualified Dublin Core, metadata conventions can be directly noted in the metadata through the use of field qualifiers. For XML-based metadata, semantic constraints and other documentation can be included in human- or machine-readable form in the DTDs or schemas referenced by the metadata.
	⇒ Standardized subject classification sources identified if used		
	⇒ Locally based vocabularies, element sets, or naming conventions described		
	⇒ Other semantics conventions documented		

10.	**Make the repository and its content known to other applications (Repository to describe)** **(Desirable)**	When users seek information, they first need to know where to search. A repository's existence and contents need to be made known to others, directly or indirectly, so that interested users and software agents can discover them.	OAI-PMH can also be used to broadcast information about the repository itself. Include a Dublin Core record describing the repository itself, along with any other relevant descriptive information, in an OAI-PMH Identify reply. Repositories can use the "friends" feature of OAI-PMH 2.0 to inform harvesters of other repositories that might be of interest.
10.1	Inform relevant gateways and registries of a repository's existence and nature (Optional)		
10.2	Inform end users explicitly of the repository's existence (Optional)		
10.3	Make end users aware of repository's content (Optional)		
10.4	Methods by which the above is accomplished (Repository to describe)		
11.	**Document the technical profile of the repository** **(Desirable)**	Learning applications that might use repositories need to know which options a particular repository has chosen for the checklist items, as well as other implementation details. If they can determine, preferably automatically, what metadata, what indexes, what identifiers, what protocols, and what policies for access and preservation a repository has, they can interoperate more effectively with these repositories.	Currently there are no standards or best practices for supporting or building repository profiles.
10.1	Repository profiles available to learning applications		

	(Optional)		
10.2	Profiles include: (Optional)		
	⇒ Metadata descriptions		
	⇒ Indexes used		
	⇒ Identifiers used		
	⇒ Protocols supported		
	⇒ Access policies		
	⇒ Preservation policies		
10.3	Profiles are machine-processable (Repository to describe) (Optional)		
10.4	Profiles are deposited in a registry (Optional)		

Standards Cited in This Checklist

The metadata, encoding, packaging, protocol, indexing, and linking standards mentioned in this report are summarized below:

Name	**Purpose**	**Reference**
ARK	Persistent identifier	http://www.cdlib.org/inside/diglib/ark/
DDI	Dataset metadata	http://www.icpsr.umich.edu/DDI/
DOI	Persistent identifier	http://www.doi.org/
Dublin Core	Descriptive metadata	http://dublincore.org/
EAD	Finding aids	http://www.loc.gov/ead/
Handle	Persistent identifier	http://www.handle.net/
IMS Content Packaging	Learning object packaging	http://www.imsproject.org/content/packaging/
IMS Metadata	Learning object metadata	http://www.imsproject.org/metadata/
Kerberos	Authentication	http://web.mit.edu/kerberos/
LDAP	Authorization, directories	IETF RFC 3377 http://www.ietf.org/rfc/rfc3377.txt
LOM	Learning object metadata	http://ltsc.ieee.org/wg12/
MARC	Bibliographic metadata	http://www.loc.gov/marc/
METS	Metadata framework	http://www.loc.gov/standards/mets/
MIME media types	Identifying formats	http://www.iana.org/assignments/media-types/
MODS	Bibliographic metadata	http://www.loc.gov/standards/mods/
MPEG-21	Metadata and packaging	ISO/IEC 21000:2004, Information technology - Multimedia framework (MPEG 21)

MPEG-21 REL	Rights Expression Language	ISO/IEC 21000-5:2004, Information technology - Multimedia framework (MPEG 21) - Part 5: Rights Expression Language
OAI (and OAI-PMH)	Metadata exposure and harvesting	http://www.openarchives.org/
ODRL	Rights Expression Language	http://odrl.net/
OKI OSIDs	Courseware interfaces	http://web.mit.edu/oki/specs/
OpenURL	Linking with citations	The OpenURL Framework for Context-Sensitive Services http://library.caltech.edu/openurl/ ANSI /NISO Z39.88
Pubcookie	Cross-institution authentication	http://www.pubcookie.org/
PURL	Persistent links	http://purl.oclc.org/
RDF	Structured metadata	http://www.w3.org/RDF/
RLI	Sharing lists of items	http://www.imsglobal.org/rli/index.cfm
RSLP Collection Description	Collection metadata	http://www.ukoln.ac.uk/metadata/rslp/
RSS	Alerting	Originated by Netscape, current control over standard unclear; see http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html **or** http://blogs.law.harvard.edu/tech/rss
SCORM	Learning object modeling	http://www.adlnet.org/
Shibboleth	Access control	http://shibboleth.internet2.edu/
SOAP	Web services	http://www.w3.org/2000/xp/Group/
SRW	Search	http://www.loc.gov/z3950/agency/zing/
TEI	Text markup and metadata	http://www.tei-c.org/
Unicode (and UTF8)	Character set (and encoding)	http://www.unicode.org/
VRA Core	Image metadata	http://www.vraweb.org/vracore3.htm
WebISO	Authentication	http://middleware.internet2.edu/webiso/
X.509	Certificates	IETF working group at http://www.ietf.org/html.charters/pkix-charter.html
XML	Structured text and data	http://www.w3.org/XML/
XrML	Rights management	http://www.xrml.org/
Z39.50	Search	http://lcweb.loc.gov/z3950/agency/