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

Appendix 2: Supporting the use of digital content in electronic learning applications.
A checklist of digital repository service requirements, with recommended best practices

John Mark Ockerbloom, Leslie Johnston, Mackenzie Smith, and William Ying

Introduction

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, and a scholar is likely to require materials drawn from multiple repositories to support research or teaching on a particular topic.

Teaching itself is increasingly supported by software applications, both to support distance education and to supplement traditional face-to-face instruction. In particular, many colleges and universities have deployed or are developing learning management systems, also known as "courseware," to support instruction. These systems are often used to deliver information drawn from internal or external digital repositories. Smaller, specialized learning applications also take advantage of content from digital repositories, such as electronic course reserve systems, personal bibliographical databases, digital portfolio managers, and presentation and analysis tools.

To make the most effective use of digital content in teaching, learning applications need to be able to easily interoperate with 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 help not only students and teachers, but it 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.

In this report, we present a checklist and discussion of digital repository services that are needed to make digital content usable by learning applications. While in practice, not all the digital information of value to scholars will be available in digital repositories, there is a set of essential services and features that any digital repository seriously intended for academic audiences must provide. We list and discuss these essential services and discuss other services and features that are desirable for interoperation with teaching and research applications. Along with discussing functional requirements for these features, we also cite current best practices and standards. We provide a table pointing to additional information on cited standards. A summary of the recommendations are provided in a succinct form for ease of use by repository operators in *Appendix 5* of this report.

This report is intended primarily for those developing repository systems and learning applications intended to work with them. More broadly, it should help those who wish to serve content to an academic and scholarly audience understand what sort of features and services they should provide to be most useful to that audience. It does not, however, recommend specific systems to buy or adopt. Many of the requirements and suggestions given here for learning applications will apply to other uses of digital repositories as well.

The authors have all been involved to various extents in the design of digital repositories or their interfaces. While our experience has been largely in academic or not-for-profit environments, we believe that commercially run repositories will also benefit from this checklist, especially when academic customers form a large part of their market.

The checklist in this report applies to use of digital content by software that supports teaching and learning, focusing on the flow of information from repository to user. It can also be useful for information to flow the other way; that is, for the results of class and other teaching activities to be deposited in repositories for archiving, or for reuse or adaptation in the future. In this report we focus on content use rather than on content deposit, but we recommend a similar checklist for content deposit as useful future work.

Readers of this report may want to consult earlier work on digital repository interoperability. The IMS Global Learning Consortium has published a digital repositories specification at <http://www.imsglobal.org/digitalrepositories/> that includes a summary of core functions and best practices for digital repository interoperability. The Open Knowledge initiative <http://web.mit.edu/oki/ > is preparing an API specification for digital repositories in learning environments, which at this writing is available in draft form to OKI partners. Additionally, CNI and IMS have written a white paper on interoperability between information and learning environments, which is now available at <http://imsglobal.org/Dlims_white_paper_publicdraft_1.pdf>.

Assumptions about process, data model, and architecture

Digital content use in scholarship is a multi-step process, involving several components other than digital repositories. As is described in more detail in the use cases in an accompanying report, the process of scholarship can be described as a three-stage process: "Gathering," where content is discovered, evaluated, and acquired for use, "Creating," where content is adapted for instructional use, or new content is created based on the information in the gathered content, and "Sharing" where the new or adapted content is then made available to others. This report focuses on the "Gathering" stage, where content is drawn from digital repositories, but the needs of the later stages are important for understanding the repository services needed at the gathering stage. Readers interested in further analysis of digital content use in scholarship may want to read John Unsworth's papers on "Scholarly Primitives". For detailed observations on how some scholars use information in an increasingly digital environment, see Brockman et al.'s "Scholarly Work in the Humanities and the Evolving Information Environment" (2001), at <http://www.clir.org/pubs/reports/pub104/contents.html>. A broader survey is described by Amy Friedlander in "Dimensions and Use of the Scholarly Information Environment" (2002), at <http://www.clir.org/pubs/reports/pub110/contents.html>.

Digital repositories can also play a significant role in the "Sharing" stage, since one way of sharing content is to deposit it in digital repositories. However, many digital repositories are only designed for public retrieval, not public deposit, and the services required for deposit are different in a number of ways from the services required for gathering information. We do not focus on deposit services in this report, but another checklist of requirements for repositories that accept contributions may be a useful future supplement to the checklist given here.

Users of digital content may carry out several activities as part of Gathering. They **discover** sources of potentially useful content. Using these sources, they **search** for content that meets their needs. From the results of these searches, they may **collect** references to relevant items they find, using information about the items to evaluate what deserves their further attention. They may **import** those items, descriptions of those items, or references to those items into learning applications. They may **save** copies of some of these items to local applications or storage. They may try to **find related** items to those they have collected, in the same or different repositories, to extend their investigations. The use cases in the accompanying report describe specific detailed examples of each of these activities.

Of the activities above, the essential activities that a digital repository must directly support are **search**, **collection**, and **import**. Discovery of information sources is clearly a prerequisite of search, but can take place outside of the repository itself. Finding related items in the same repository can be thought of as a special case of search. Saving items can be thought of as a particularly useful type of import, one that is not always feasible for information with challenging space, processing, or access control requirements. Importing items in a way that allows them to be viewed by students may be sufficient for instructional purposes, even if the items cannot be saved in full outside the repository.

Our description of the activities above assumes a simple data model for digital content. We assume that there exist distinct, identifiable pieces of digital content, which we call **items**, that can be searched for, collected, and imported for instructional purposes. An item may be something used all at once, like a research paper, or something that is only used in part, such as a large dataset. Items can be found by searching **collections**, groupings of items that can be addressed and queried through a common interface or set of services. Items, and possibly collections as well, have **metadata** associated with them, information that describes them and otherwise aids in their use and management.

The component architecture underlying this activity is also straightforward, and is illustrated in Figure 1. Digital **repositories** store items of digital information, and provide interfaces to services for accessing this information that should fulfill the essential requirements we give in our checklist. We assume that repositories provide content, not simply metadata, to users. Examples of repositories include ArXiv.org (for scientific papers), the Library of Congress' American Memory project (for text, image, sound, and video), and the ICPSR data repository (for data sets).

Teachers and learners use the content of digital repositories through **learning applications**, programs that find and present digital content for use in teaching and learning. These applications may be broad courseware suites, or simple generic retrieval or display programs, or something in between these extremes. Typically, uses of digital content in teaching and learning involve some form of software mediation beyond whatever user interface the repository itself might provide. Examples of applications include **courseware** packages (such as Blackboard), **bibliographic managers** (such as EndNote), and **presentation and analysis software** (such as Insight or SPSS).

Due to the wide selection and range of interfaces of repositories, there has also emerged a layer of mediators between repositories and applications, which we call **gateways**, Gateways help users locate content they need in appropriate repositories, and from the point of view of end users and their applications, may provide many of the same services as repositories. However, they do not have primary responsibility either for storing that content or for presenting it in an online learning context. Search engines, indexes, and portals are often best thought of as gateways. Repositories of metadata, without other digital content, can also be considered gateways for the content described by the metadata. A typical gateway mediates access to many repositories. Examples of gateways include PubMed and the Gateway to Educational Materials. Even a general purpose search engine like Google can be thought of as a gateway, as it mediates access to many publicly accessible repositories as well as a vast array of ordinary web sites.

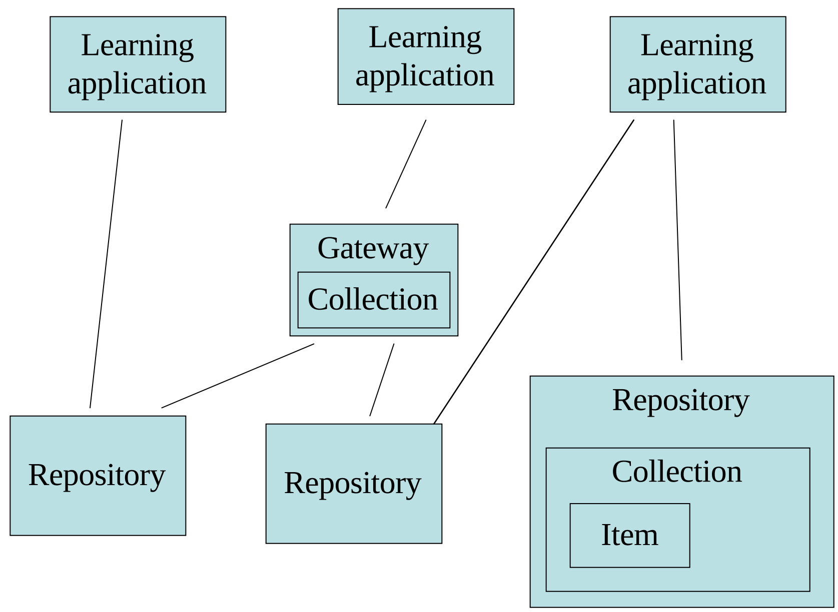

Figure 1: Learning applications search for, collect, and import items contained in collections managed by repositories. Applications can also use gateways that mediate access to virtual collections of items from multiple repositories.

General design principles

We note a few general design principles that repository services should follow in order to be accessible in useful ways from learning applications.

Ensure broad accessibility of the repository. In contrast to the positive requirements of our checklist, this is largely a negative requirement: **do not** implement interfaces that will prevent applications from making your repository material accessible. For example, do not make access to items dependent on a proprietary protocol only supported by one commercial vendor, unless you want to limit your audience to that vendor's customers. Do not mandate browsing through non-textual means, such as an image map, if you need to conform to section 508 of the United States federal disability regulations.

The positive aspects of this requirement mainly have to do with following commonly used standards for interoperation, including standardized data and metadata formats, protocols, linking, and indexing standards where possible. Details on current best practices in these areas will be given in the discussion. One important standard that cuts across repository services is character coding standard. We recommend using character encodings that are compatible with a wide range of languages that may be represented by a repository's content and users. The Unicode character set, which covers nearly all the world's major languages, is now widely adopted, and its UTF-8 character encoding is one of its more commonly used byte representations, and is backward compatible with ASCII, the most widely recognized character set on the Internet. To support multilingual user interfaces, repositories may need to standardize their own output formats, and use standard protocols that can be adapted to a localized end user interface by a learning application or gateway.

Another increasingly important standard, particularly for metadata, is XML, a general format for marking up structured text and data. Many of the metadata standards discussed in this report are based on XML. XML makes it easy for new metadata and content structures, or extensions of existing structures, to be defined and parsed. A growing array of tools is available to create and parse XML, and translate between XML and other common formats. XML builds on Unicode, using it as its base character encoding.

Provide access control to items that does not hinder learning applications. While some digital repositories may expose all of their content for the world to see, many repositories, particularly those of commercial publishers, may limit what users can do with their content. They may also only provide access to the content to trusted users or paid subscribers.

The decision to impose access control, and the choice of terms of use, are up to the repository owners. However, the access controls should still make available to authorized users the essential services we give in our checklist. Some repositories have one set of access terms that apply to their complete collection. Others may need to provide a wider range of finer-grain controls. For example, some items in a repository might be completely accessible to the public, while others might have their metadata viewable by all, but their contents only viewable by authenticated users at a subscribing institution.

Whatever policies a repository decides to use, following certain usage and technical standards can make a repository more easily interoperable with learning applications. A repository meant to be accessible to the general public or general gateways and tools should delay authentication until the actual point of need. It should minimize the number of times a user has to authenticate, through session state or by using credentials from earlier authentication. If possible, repositories should avoid requiring registration of individual users that is separate from pre-existing institutional user ID schemes and systems.

Access controls should be documented, preferably directly in the administrative metadata of items or collections. Not only does this make it clear to users what they (and their colleagues and students) can do with items, but some repositories might be able to automatically enforce applicable access rights encoded in the metadata. Such metadata also allows users to search based on rights criteria (such as by specifying, for example, that one is only interested in items that are free to access).

Authentication regimes now in widespread use may be easier to integrate with a diverse set of learning applications and subscribers than more specialized methods. For example, many licensed materials simply restrict usage of their collections to certain IP ranges, set to cover the address space of subscribing institutions. For finer-grain access control, individual or group logons that relate to privileges and can be mapped to collections and objects may be necessary. Session state should be temporarily stored to retain authentication state and minimize the need for re-authentication. Relevant technologies for authentication and authorization include the following:

- Kerberos
- LDAP
- Proxy servers.
- Public Key (X.509) certificates.
- Virtual Private Networks (VPNs).
- Institutional single sign-on services (e.g. WebISO, Pubcookie)
- Shibboleth is an important technology to watch, though not yet mature.

Checklist of services and features

To be useful for online learning applications, sources of digital content must provide these five essential repository services:

Support search for items.
Provide stable references to items.
Provide ways to get and use item content.
Provide standard or documented metadata for items.
Document policies and functions of the repository.

The first three items above directly enable searching, collecting, and importing. The remaining two provide essential information that supports these and other activities. Metadata provides crucial information for searching, helps users evaluate what items they should collect, 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.

For ease of discussion, we divide the full list of recommended services and features into categories, and underline the essential ones.

Finding content:

<u>1. Support search for items.</u>
<u>2. Provide standard or documented metadata for items.</u>
3. Support search via software agents.

Collecting content:

<u>4. Provide stable references to items.</u>
5. Support citations (in recognized scholarly formats) for items.

Accessing content:

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

Documentation:

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

We have not included on this list features related to depositing items into a repository, but instead focus on the use of items **from** a repository by learning applications. After some consideration, we also omitted features such as versioning support, usage statistics, or refinement of search results. 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.

Below we discuss each checklist item in detail. A checklist form summarizing the discussion can be found at the end of this report.

Finding content:

<u>1. Support search for items (Essential feature)</u>

What and why: In order to use content from repositories, teachers and learners have to find what they need. Some method for locating desired content, therefore, is essential.

There are several methods by which users can find desirable content. They can **query** for content that matches criteria they specify. They can **browse** to discover what a repository contains, and explore its contents along various dimensions, such as title, author, content type, and subject. They can **follow relations** between items to learn the wider context of content, and find items related to content they already have found. And with the right infrastructure and setup, they can also **be alerted** to items that may interest them, even if they do not actively search for them.

<u>Essential functionality</u>: Query and browse support is necessary because repositories are likely to be large, and users of learning systems are not likely to have exhaustive knowledge of their contents. Standalone courseware systems, where all course content placed in a small "class readings" folder may be usable just with browse, but even there materials meant to be used beyond the scope of a single course or unit could benefit from query capability.

Users must be able to query for items based on essential descriptive metadata. In particular, items should be locatable by their title, if one exists. Search by author or creator is also essential in repositories that contain content attributable to particular individuals or organizations.

Users must be able to determine what is in a repository, through browsing. Users must be able to inventory all items in the repository; and if there are multiple collections, inventorying should list all collections and have the ability to enumerate those collections' contents as individual subsets. Inventories must be available both to end users browsing the contents of a repository, and to programs that can index, list, or harvest the repository.

Desirable functionality: Title, author, and subject, when available, are the most desirable metadata fields to search on; date is highly useful as well. Ideally, any information one might use to discover an item should be searchable. Users should therefore be able to search based on any descriptive metadata for items. This should include metadata that can be used to locate the most desirable items, even if this metadata is not labeled “descriptive” in the repository’s metadata schema. For example, format information may be kept in a repository’s “administrative” or "technical" metadata, but still be useful for users that wish to limit their search to items in particular formats.

Queries: For queries, it is useful to allow queries both on particular metadata fields, and general keyword queries. While the former is more precise, the latter is much simpler, and in practice can make it much easier for new users, or any user who cannot take the time to adapt a query to a repository’s particular organizational scheme, to find items that may be of interest.

Queries based on content, and not just metadata, can be highly useful. Some repositories (such as Greenstone and BePress) implement full text search natively. Content-based search of non-textual information is less mature than text search, but some noteworthy attempts include Google’s image search (which uses nearby text or file names as clues for guessing the content of images), and Informedia’s video search (which uses captioning and other cues to find relevant video clips).

Browsing: Users should be able to browse the contents of a repository by meaningful categories. Title and creator browsing are the highest priority stated by those seeking to browse. Date-based browsing is also often desired. Date of publication can be useful for historical repositories, as well as for keeping track of successive versions of documents. Date of accession can be useful for users who want to see what has been added since a previous visit.

Other useful browsing dimensions include format (all images, texts, video, etc.) and subject. Subject is by far the most problematic, as it requires the normalization of subject headings applied through disparate community-based encoding standards, as well as the augmentation of subject metadata for objects acquired from sources outside the unit managing the repository. Hierarchical browsing can be useful in collections with parent-child relationships and ordering. Some subject taxonomies particularly lend themselves to hierarchical browsing.

Finding related items: Repositories should enable users to find items related to ones that they find, to give context to what they find and extend their investigations. This can be done in various ways. Metadata included with an item can be linked to a query that finds more items in the repository with similar metadata (such as links to items with the same author or subject terms, or to alternate versions of the same document). Some repositories go further, analyzing their usage history of contents to provide links to other relevant items. In the commercial world, Amazon uses purchasing histories to link “related titles” that customers have bought together in the past. In the scientific world, repositories that track works that cite other works can be very useful.

For links to related information outside a repository, repository interfaces can activate URLs or other links in item metadata (or content). If bibliographic citations appear in a repository, they can be mapped into OpenURLs to enable linking to external resources via the user's local OpenURL resolver. Determining the appropriate resolver to use for a given user is not always straightforward, but commercial vendors have found solutions to this, using things like cookies, IP addresses, or user profiling.

Alerting: Frequent users of digital content may need to know when new material is available relevant to their interests. Alerting criteria can be specified explicitly by a user, or implicitly by the user's behavior. User interface challenges for alerting services include keeping alerts focused enough to maintain a user's interest, and making notification neither too inconspicuous nor too intrusive.

Search results: The results of a search should be presented in a way that helps users select the material they want. For queries, relevancy-based ranking is a highly valued feature of Internet search engines. When browsing, items are usually shown sorted in an appropriate order for the aspect being browsed, such as by title or by author. Sorting of search results by other criteria, such as date, can be highly useful for different types of search. The display of results can also be optimized to their format and expected use. Searches of text documents, for instance, can show short summaries or relevant excerpts; searches of images can present thumbnails. The visual layout of image results can encourage browsing in multiple dimensions.

Where it fits in the architecture: In most cases, repositories will provide their own search, or delegate it to a closely coupled gateway. Native search can be more precise and informative than generic searches provided by general-purpose gateways, both because repositories have information not available to the gateways, and because they can optimize the search function to the strengths of their particular collections.

However, it is also possible to provide search by exporting metadata or content to an external gateway, which then provides search for the repository and others. If necessary, this can substitute for native search, as long as users know how to find and use the appropriate gateway. Ideally, a repository should support both options. (See the "support search via software agents" discussion for more details.)

Technical recommendations: A wide variety of full text indexing and search programs now exist, and we recommend full-text search for text-oriented repositories. For image search, we recommend thumbnail support, which lets users spot images quickly. For all media, flexible metadata support is important, since different collections may require different kinds of metadata. For supporting search through gateways, OAI-PMH is an increasingly common way of exporting metadata. Public Internet search engines can index and provide text-based search for repositories that are open-access and can be traversed by simple hyperlinks. They are not guaranteed to index all of a given site, or do so in a timely fashion, but some search engine companies are offering special services or appliances to provide complete indexing of local repositories.

Browsing is essentially a rules-based extension of search functionality. If one does not wish to provide a special browsing interface, a series of queries can be pre-set to return a complete object-level inventory; a list of collections; object-level inventories for individual collections; inventories sorted by creator, title, date or subject; all objects added since a specified date; objects by MIME type; and so on.

2. Provide standard, documented metadata for items (Essential feature)

What and Why: The most basic requirement for a repository in addition to supporting discovery and viewing of its contents is supplying the item-level metadata that describes them. Such metadata helps users find appropriate content, and understand the nature of the content they find. In order to be understood by users and applications, repositories should provide metadata in standard formats, or document the conventions used for its metadata, or both.

Essential functionality: Machine-readable metadata is essential, so that other applications can index, translate, and display it. The repository must also present metadata in a form that end users can read and understand. The minimum item-level descriptive metadata is a title. The minimum item-level technical metadata for displayed or exported content is a MIME media type.

Desirable functionality: Basic descriptive, technical, and administrative metadata should be provided. Identifier, Creator, Date, Type, Format, and Rights elements are strongly recommended in addition to a Title. For use in teaching and research, sufficient metadata should be provided to make it possible to cite an item in scholarly form. Citation support is discussed in more detail in a later item. Item metadata is preferably provided as key-value pairs.

Including structural metadata is important for large items, or those with complex structure, to be viewed and navigated in intelligible ways. Viewing and navigation of complex items is discussed in more detail under the checklist item "Provide views of item content."

A repository intended for educational use should support the inclusion of item-level descriptive metadata to document the purpose, applicability, educational goals, and prerequisites of its content. Such metadata can greatly assist an instructor in finding educationally suitable content. It should be available via searching and browsing. Repositories that ingest educational materials may usefully provide a mechanism for the creation of such metadata at the time of item creation or repository ingest.

Where it fits in the architecture: Basic descriptive, technical, and administrative metadata should be managed by the repository. It should be provided through search, browse, and exposure of machine-readable data.

Technical recommendations: Dublin Core (DC) is the most ubiquitous standard that can be recommended as the minimal set of metadata elements that repositories should provide. The Dublin Core Library Application Profile <http://dublincore.org/documents/library-application-profile/> should be considered; at a minimum, unqualified OAI-Dublin Core <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm#dublincore> is recommended. Repositories need not contain DC records natively, but they should provide metadata in a format that can be mapped to DC records, and ideally provide either OAI encodings or support for harvesting via OAI. The repository should be capable of displaying a Dublin Core record or its equivalent at an item level for users.

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, and DDI for data sets. The same standards, applied in conjunction with METS, should be considered for fuller administrative and technical metadata.

In the domain of e-learning systems there is consensus (if not a large body of practice) around a family of descriptive standards for learning objects developed by the IMS Global Learning organization <http://www.imsglobal.org/>. **Learning objects** are items of digital content that have been specially packaged for online teaching and learning, and typically managed by applications and repositories built for that purpose. They can be created from general-purpose repository items, though, if adequate metadata exists for them.

The IMS metadata specifications for learning objects are derived from the IEEE LOM (Learning Object Metadata) standard and are consistent with similar standards from other education and training organizations like SCORM. The IMS metadata specification provides a set of descriptive elements that can be applied to learning objects of all kinds, and XML and RDF bindings for encoding of these elements. These elements support the description of educational suitability that we recommend above. It is possible to create Dublin Core records from IMS metadata and many other metadata schemas. Conversion the other way is often possible in theory, but in practice automatic conversion to IMS metadata often yields extremely minimal or imprecise records, due to the minimal semantics defined for unqualified Dublin Core.

Beyond descriptive metadata, repositories should consider how to handle administrative, technical, rights, and structural metadata. Standards for these are less well developed, but there are some that are emerging for the e-learning domain.

Administrative: This category includes basic information about the provenance and current stewardship of an item of content. While there are not standards or best practices for this type of metadata, it might include the name and contact information of the owning repository, and policy statements about the level of commitment of the repository for long-term preservation and availability.

Technical: Technical metadata standards are generally format-specific, since they are used primary for learning object life cycle management and long-term preservation. There are emerging standards for some common technical formats (e.g. for still images) but not for the majority of formats in use. Repositories should consider capturing some basic metadata for learning objects that can be extended as standards become available. For example, DSpace captures a MIME type, source filename, file size, submission date, and MD5 checksum for each bit-stream submitted to it.

Detailed technical capture information should be recorded for all media files, although use of such capture information by a learning application would likely require some prior arrangement, so that application could configure itself to take advantage of such additional data in presentation.

Rights: The two dominant rights expression languages in development now are XrML <http://www.xrml.org/> and ODRL <http://www.odrl.net/>. The former is documented but proprietary. In particular, it is patented and must be licensed for use by digital rights management applications. The latter is an open standard without encumbrance and is widely favored by the e-learning community.

Structural: Like other types of digital objects, learning objects can be complex, composed of multiple parts in particular structures. For example, a thesis might consist of 200 individual TIFF images in chapter and page order, or a learning module might consist of several units that must be completed in sequence. Standards for showing specific relationships between parts of an object are beginning to emerge in practice.

In the library domain, METS, which includes structural as well as other kinds of metadata, is used by a growing number of digital library systems. Some repositories have used it to support hierarchical browsing, navigation of complex objects, and linking to related items. For example, the RLG Cultural Materials collection contains images organized into a hierarchy of groups to represent their relationships, and also supports the inclusion of related text, audio, or video. RLG provides a METS viewer for accessing the repository. The METS viewer lets users increase an object's size and examine descriptive data and any inventory of the digitized objects associated with a particular work. It also allows page turning and viewing of related items.

In the e-learning domain the IMS Content Package (IMS CP) is becoming the norm. Crosswalks are being developed between IMS CP and METS to allow repositories that support one of these to interoperate with other domains, so while a given repository could choose to support either METS or IMS CP by default, it should be able to translate into the other for cross-domain use. One other standard that is beginning to emerge in this area, especially for digital video and multimedia material, is MPEG-21, though it is not yet clear whether it will be widely adopted by digital libraries or e-learning repositories.

Most of the metadata schemes mentioned above are encoded in XML, or can be easily translated to an XML encoding (e.g. MARC to MARCXML). We strongly recommend that repositories be able to ingest and export metadata in standard, serialized XML format, as this is likely to be compatible with a wide variety of gathering and depositing applications. Repositories may of course store metadata differently internally for optimization.

Challenges and open questions: Provenance metadata, which tracks the history of content, becomes increasingly important as content is aggregated, modified, and shared. Among other things, they help users track the origin, reliability, and rights of digital content. There is not yet a generally agreed upon way of representing provenance in digital object metadata, but it is the subject of ongoing research. Systems that support editing histories, such as source code control systems like CVS, or group editing systems like Wikis, capture some provenance information, but there is not yet a widely recognized standard form for provenance information generally.

Providing detailed machine-readable format metadata is another ongoing challenge. MIME types are often not precise enough to fully identify file formats. More detailed format specifications and profiles could contain information such as DPI, pixel dimensions, and TIFF profile identifiers for images, or specific schemas for XML metadata. The Digital Library Federation has proposed a digital format registry where richer technical format specifications could be recorded, and a format-encoding vocabulary can be specified. The METS community is looking at best practices for the encoding of technical metadata.

3. Support search via software agents:

What and why: Since relevant items can be in any of hundreds of repositories, it is often not practical for users to individually search each of them anew every time they look for information. Repositories should support more efficient ways of getting needed information to interested users. Letting programs as well as users search repositories supports the use of software agents that can let users find information much more efficiently than they could by searching them manually one at a time.

Desirable functionality: Repositories should be searchable through standard search protocols, through standardized authentication schemes. Gateways and applications that support these protocols can then use them to support new kinds of searches that cover many repositories through a single user interface and display relevant results from many repositories at once. Meta-searching gateways, which effectively search many repositories at once, can increase the efficiency of user searching. Flexible machine interfaces for new and canned searches allow applications to provide targeted search and alerting services to teachers and students, informing them of new items that are of interest to them.

Additionally, allowing repository metadata to be harvested lets gateways include multiple repositories in their search without the overhead of querying the repository every time a user wants to search that repository's contents. It may support alerting services. Harvesting is the primary method by which public Internet search engines like Google make it possible to search large portions of the Internet at once.

Where it fits in the architecture: Multi-repository search services are usually provided at a level above the repository, either by a gateway or by an application. Repository interfaces play important roles in supporting these services. Meta-search typically requires either support for common search protocols, or some method for gateways to harvest metadata and other information used in a federated search. Portal and alert services may need interfaces for getting dynamic results of a canned search, extracting metadata on the most recently added items meeting certain search criteria, or allowing user-directed search on a pre-selected collection of items.

In some cases, the repository itself might provide portal features. For example, DSpace provides an alerting service for new items in its collections matching previously expressed user preferences. Alerts maybe a useful application for a repository to provide itself, when the alerts involve information or criteria that would not be known or supported by a more general-purpose portal or gateway, or when a repository owner wants to promote its own content. However, users might not want to sign up for separate alerts for dozens of repositories they might be interested in, so some way of feeding a more general alert system is desirable.

Technical recommendations: Z39.50 is the most widely supported searching protocol 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. It is not yet as firmly established as Z39.50.

For harvesting, OAI-PMH is an important protocol. It allows metadata to be retrieved for all or a selection of items, including selectively harvesting the newest items in a collection. Harvested metadata can be aggregated to provide the core data for meta-search, and alert services can be built to notify users of newly harvested resources. 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. Most general-purpose OAI harvesters today focus on Dublin Core metadata. 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.

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 feature)

What and why: Users of learning applications need to know how they can reliably get to selected content in a repository. Scholarly users that publish papers will also need to use and refer to specific content inside a repository.

Essential functionality: At a minimum, the repository must provide a stable identifier for each item in the repository. It must be possible for external systems to use this identifier to locate the item in the repository for as long as the item exists in the repository.

Desirable functionality: Beyond the basic stability requirements, we recommend that identifiers also be unique (that is, not used for any other item, even in other repositories), and persistent (that is, capable of outliving an item's original repository). For some content, permanent archives may be desirable. The requirements of permanent archival systems go well beyond the scope of this report, but have been discussed in detail elsewhere. See, for example, the National Digital Information Infrastructure and Preservation Program of the Library of Congress, at http://www.digitalpreservation.gov/.

It is also useful for repositories to be potential targets for OpenURL resolution. This would require a method for referring to items based on citation information.

In order to publicize and encourage the use of stable reference, a repository's search interface should include conspicuous stable URLs for items that users find. Stable references should be included in citation downloads, if a repository offers that service.

Where it fits in the architecture/Technical Recommendations: Generally, stable identifiers need to be supported in the repository itself. Underlying technology for such 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, are also being developed and evaluated by some major repositories. Debate continues over which of these 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.

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.

There is a key distinction between linking directly to content, and linking to an item record, which gives access both to content and metadata. 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.

5. Support citations (in recognized scholarly formats) for items

What and why: A repository should support the creation or export of citations in recognized scholarly formats for items, based on their descriptive metadata. This capability helps users systematically collect and manage citations and bibliographic data for their own papers and publications.

Desired functionality: As a mimimum, a repository should provide a text citation that can be easily copied and pasted. It is also useful for metadata to be exportable to a saved citations list, or directly to bibliographic software such as EndNote, ProCite, Reference Manager, RefWorks, or spreadsheet software such as Microsoft Excel.

Citations should include persistent identifiers, if available. For exporting metadata for citation purposes, a thumbnail would be desirable for image-based content.

Technical recommendations: As mentioned above, citations to digital items require robust stable identifiers. Multiple technical formats for citations may also be necessary. For example, JSTOR has implemented a citation manager with the following functionalities and formats:

- The **printer-friendly** format is 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.

- The **citation-manager** format outputs each citation into a machine-readable format that consists of 22 tags. This format is designed to be imported into bibliographic software packages, such as EndNote, ProCite, Reference Manager, and RefWorks. Filters for importing citations into a bibliographic manager are provided.

- The **tab-delimited** format is a simple text file with field values separated by the tab character. This format can be used to import citations into a spreadsheet application (such as Microsoft Excel).

Challenges and open questions: Agreement on citation formats across disciplines will not be simple. Once agreed upon, a list of item level metadata elements that are essential for citation will likely be required to support citation construction. Normalizing metadata elements across data types from myriad sources is not an easy task. IMS is now developing a Resources List Interoperability (RLI) standard that may eventually standardize the creation and exchange of citation and other bibliographic data in the e-learning context.

Accessing content:

6. Provide ways to access and use content (Essential feature)

What and why: 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. This is the heart of the "import" activity described earlier.

Essential functionality: Users must have sufficient access to content to allow its use in teaching and scholarship. The content need not be unrestricted or freely available, but basic views must be available to anyone with the proper authorization. Users must be able to retrieve the actual item content and then process it further, or the repository must provide views of that content that users can view, navigate, and analyze appropriately. These two alternatives are described in more detail under subsequent checklist items.

Desirable functionality: Selective access to content may be desirable for certain types of content. 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"). Large audio or video recordings might be usefully accessed in selected snippets.

Technical Recommendations: 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. It includes operations for clients to receive disseminations on objects in the repository and to discover information about an object using object reflection. The Open Knowledge Initiative (OKI) is defining a Content Repository API to fulfill some of these functions. SOAP based web services are recommended to interoperate with learning applications that tightly integrate content inside the repository with their authoring tools, for those repositories that do not allow download of contents.

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.

7. Provide views of item content

What and why: 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.

Desirable functionality: Repositories should provide a way for content to be viewed via a web browser. For some content, this might require translation of the content to HTML or another format widely supported in Web browsers, or providing a plugin or applet to view the content. Along with a basic view of the content, it is useful for repositories to display metadata, links to an item's underlying files or bit-streams, and other potentially useful disseminations. Repositories should provide administrative metadata where needed, such as date of creation, collections an item belongs to, and copyright information for content.

Repositories should provide some default view of an item if a complete export of the item is impractical. For example, a high-resolution TIFF that is too large to export to a browser (or too valuable for a publisher to export in full detail) might be represented by a screen-sized JPEG or MrSid image. In other cases, a simple link to the underlying bit-stream asset will be acceptable; for example, a PDF file can just be exported as-is to a user's browser plug-in. If an item is textual, display of the full text should be offered.

Repositories may also need to support navigation within complex items stored in the repository. Examples of such navigation include hyperlinked tables of contents, hierarchy traversal, page turning, and image zooming. If repositories do not provide this functionality natively, they may need to allow the items to be exported with sufficient structural data or metadata to allow other applications to support such navigation.

Technical recommendations: 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.

Object-oriented repository architectures such as Fedora can support mapping of objects to behaviors and disseminators to present different views of various content types, which 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. Fedora supports the ingestion of complex digital objects through METS to present them to users in a hierarchical or sequential manner once discovered.

Sophisticated data modeling capability is required to support parent-child or sibling navigation. Repositories providing such navigation natively will need tools to manage the hierarchies. OKI has a Hierarchy OSID (API) that manages parent-child relationships among elements. In addition to simple tree structures, the OSID supports hierarchies that are recursive and have nodes with multiple parents, enabling simple hierarchy browsing capabilities.

Challenges and open questions: Repositories do not necessarily know how "smart" a user's display application is. If repositories contain a wide variety of data, a simple browser client might not be capable of displaying them all in a complete or consistent manner. This is less of a problem if repositories follow well-known standards for content and display formats, and use formats that are well documented.

Making the exported or printed version of an item consistent with its display in a repository viewer can be a challenge, for reasons similar to the above.

8. Allow items to be copied into local systems

What and why: Instructional materials can be much more effective when teachers can adapt them to their own learner's needs, reuse them in different learning contexts, and distribute them in ways that are most convenient for their students. To make these things practical, item content should be exportable into local systems for reuse and modification. Even for simple presentation, downloading items to applications like PowerPoint (still the most commonly used software for classroom presentations) is often necessary. Downloading content is a necessity for offline viewing and presentation, and can be a practical necessity for many types of specialized applications. We realize that instructors sometimes must use content from repositories that do not allow download, so we cannot consider this an absolutely essential feature. But it is highly desirable.

Desired functionality: Repositories should allow users to download selected content into their local applications or file systems. Ideally, users should be able to get all metadata, along with all content bit-streams that are associated with the item. To protect intellectual property or minimize the load on repositories, though, some content may be downgraded to lesser resolution for export, and some repositories may limit the number or rate of downloads that are allowed. Repositories might suppress internal administrative or version data if that is not of interest to learning applications.

Technical Recommendations: Packaging standards for learning objects and other repository items can use many of the same standards that are used to record structural metadata. The same options of METS, IMS Content Packaging, and MPEG21 are available to package learning objects, simple or complex, together with their metadata for exchange with learning applications or other repositories.

Challenges and open questions: The biggest challenges are related to intellectual property rights issues, specifically with documenting rights and restrictions and enforcing them properly.

Some types of content are not easily exportable due to the lack of widely adopted standards for packaging or even modeling objects, or because the content is tightly bound to the interface that provides the content. Consider spatial data objects. Even if items are exportable, their packaging formats may not give sufficient semantic information to ensure that an application can understand the organization or use of the items. The Digital Library Federation's Distributed Open Digital Library project (DODL), which promotes deep sharing of collection content, may assist in the development of object models and packaging standards.

Most repositories support basic bit-stream formats that may or may not conform to a standard file format or MIME type. In such cases, users may have to decide themselves how to process downloaded items. The Digital Library Federation's Format Registry project may in the future provide access to format information and services that go beyond the capabilities of MIME.

Without full download capability, a repository may have sole responsibility for many of the features in our checklist that might otherwise be delegated to learning applications, gateways, or other agents.

Documentation:

9. Document policies and functions of the repository (Essential feature)

What and Why: It is essential for repository rights, restrictions, functions, and critical policies 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.

Essential functionality: Critical policies include copyrights and related rights, security, and privacy. Those rights and restrictions are sometimes implicit in the access control. As an example, most publisher sites, while not providing detailed information on the rights for each item, at least state somewhere that a subscription is required, and give terms of subscription and use to those who ask about it. If the repository does not use standard metadata, it must document its metadata conventions, so that users and applications understand how to interpret the metadata.

Desirable functionality: It may sometimes be appropriate to document access rights at item level.

Ideally, metadata documentation should be provided even if standard metadata formats are used, to note semantic conventions. For example, if standardized subject classifications have been assigned, the source of the controlled vocabulary or vocabularies should be identified. If locally based vocabularies, element sets, or naming conventions are used, they should be described.

Some systems, such as DSpace, advise content contributors of the preservation support levels they can expect for the files they submit. This policy can give users assurance about the future usability of contributed items.

Where it fits in the architecture: Documentation of a repository's functions and policies should be supplied by the repository itself. A registry of repository information might serve as an alternate source of this documentation. At the moment there is no authoritative registry of this type, but the OAI community has some informal registries of information on known OAI data providers.

Technical recommendations: At the most basic level, an "Identify" call to the OAI Provider front-end on the repository supplies basic repository documentation. However, 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 may do well to follow the recommendations given in RLG/OCLC's paper on trusted digital repositories <http://www.rlg.org/pr/pr2002-repositories.html>.

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. Human-readable documentation is especially important for repository-specific conventions.

10. Make the repository, and its content, known to other applications

What and why: 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 can discover them.

Desirable functionality: Relevant gateways and repository registries need to be informed of a repository's existence and nature. End users need to be able to find material in that repository, but do not necessarily have to be informed explicitly of the repository's existence.

Where it fits in the architecture: Much repository publicity is done by humans, rather than machines, and certainly informing relevant user communities and gateways is important. But repositories themselves can inform peers, gateways, and registries about themselves, as described below.

Technical recommendations: OAI-PMH is currently the most common method to broadcast information about a repository's content, and it can also be used to broadcast information about the repository itself. We recommend including 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.

11. Document the technical profile of the repository

What and Why: Throughout this report, we present different options for the implementation details of each checklist item. We often cannot be more prescriptive, either because different options are appropriate for different contexts, or because none of the options have been conclusively demonstrated to be superior to the others. Learning applications that might use repositories need to know which options a particular repository has chosen, 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. **Repository profiles** including this information are thus highly desirable.

Currently there are no standards or best practices for supporting or building repository profiles, so we cannot recommend a particular convention or format. Making much stricter recommendations for particular repository standards and protocols could replace the need for profiles, but we know that to be impractical. Instead we recommend that further work be commissioned to develop specifications for such profiles.

Desirable features: Specific information about a repository's implementation profile should include supported content types, metadata, indexes, protocols, identifier formats, and access policies.

Where it fits in the architecture: A repository registry could be a useful neutral ground for specifying and managing repository profiles in standard forms. Such a registry could also be used to discover new repositories. Repositories can also supply their own technical profiling information, though, and in the absence of repository registries, this is the only practical option.

Conclusion

As the domains of libraries, digital repositories, and learning applications begin to collaborate and cooperate more, having a common vision of the functionality teachers and learners need from the content available to them is becoming increasingly important and more achievable. Once a body of such repositories of learning objects exists, elaboration and refinement of these guidelines will be possible. In the meantime these recommendations are based on current best practice in the digital library domain.

Digital object repositories designed by and for libraries have long supported many of the functions and standards described in this checklist, but those designed to store and manage learning objects for learning applications have often implemented a much simpler set of functions and therefore cannot support the wider world of digital content available for e-learning. The recommendations in this report include a set of basic, **essential** functions which repositories supporting e-learning should all be assumed to support, and large number of **optional**, but highly desirable, functions which will improve the ability of external learning applications to leverage and exploit the learning objects. The checklist represents the collective experience of many years of practice in running digital repositories, as well as a snapshot of current standards and best (or common) practices for implementing many of the functions.

The repository requirements for persistent identification, discovery, searching and browsing of appropriate metadata, retrieval (through persistent, remote linking or direct provision), and policy documentation are not onerous to expect of professionally managed repositories. Using the standards and best practices described will help ensure interoperability with a range of learning applications and the ability to move content between repositories when appropriate. Over time, many of the optional functions may become so trivial to implement and commonly found that they move into the essential category.

Acknowledgements

We wish to thank Dale Flecker and the members of the repository interoperation group that he led, for many of the recommendations in this report, as well as for many useful suggestions for revision. Thanks also go to the Andrew W. Mellon Foundation and the Digital Library Federation for convening and sponsoring the meetings of this group. Special thanks go to David Millman for his useful suggestions for recent drafts.

Standards Cited in This Report

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	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	http://www.chiragione.org/mpeg/standards/mpeg-21/mpeg-21.htm

OAI (and OAI-PMH)	Metadata exposure and harvesting	http://www.openarchives.org/
ODRL	Rights management	http://odrl.net/
OKI OSIDs	Courseware interfaces	http://web.mit.edu/oki/specs/
OpenURL	Linking with citations	http://library.caltech.edu/openurl/
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/workinprogress.cfm
RSLP Collection Description	Collection metadata	http://www.ukoln.ac.uk/metadata/rslp/
RSS	Alerting	Control over standard not clear; see http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html
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/