DIGITAL LIBRARY
FEDERATION

Digital Library Content and Course Management Systems: Issues of Interoperation

Report of a study group funded by the Andrew W. Mellon Foundation

Co-Chairs: Dale Flecker, Associate Director for Planning & Systems, Harvard University Library, and Neil McLean, Director, IMS Australia

July, 2004

Copyright 2004 by the Digital Library Federation.

Table of Contents

Executive Summary

With funding from the Andrew W. Mellon Foundation, an *ad hoc* group of digital librarians, course management system developers, and publishers met under the aegis of the Digital Library Federation to discuss the issues related to the use of digital library content in course management systems. The size, heterogeneity, and complexity of the current information landscape create enormous challenges for the interoperation of information repositories and systems that support course instruction. The group has created a checklist of things that operators of digital content repositories can do to help ameliorate the complexities of such interoperation. It also explored through the means of use cases the utility of tools which help instructors gather information resources from various distributed information repositories for teaching purposes, and created a model of how the group envisions the interaction of users, tools, and information repositories in the future. Understanding the complexities of the information landscape, and the importance of tools to simplify interactions with that landscape, is critical for those building systems and services in this domain. The group believes that it is now important that the community move from theoretical discussions of interoperation of content repositories and instructional systems to real-world demonstration projects in order to further our collective understanding of the needs of users and the realities of systems interoperation.

Introduction

American institutions of higher education today are awash with digital information resources. Members of the educational community commonly have access to thousands and thousands of electronic books and journals, hundreds of digital reference works, increasingly rich collections of digital pictures, videos, and music, and large databases of survey, geographic, and scientific data. Few areas of academic work are not dependent on at least some digital resources at this point, and the range and importance of what is available continues to grow dramatically

While many digital resources are maintained and accessed through the local environments of scholars and research groups, a very significant number, particularly materials of wider interest, are captured in the more formal systems of publishers, digital libraries, and institutional repositories.

In the same period that the range and scale of digital resources available within universities was beginning to grow dramatically, so was the use of information technology tools to assist in or augment teaching and learning. These tools range from the small and personal (personal web sites, PowerPoint) to large-scale institutional course management systems. Given the richness of digital resources available, one might have expected that educational tools would quickly become a significant vehicle for providing students with access to digital library resources relevant to their courses. However, there is a wide-spread perception that the level of integration of digital materials from formal repositories in educational systems remains relatively low.

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. The issue has been addressed on a technical architecture level in the "repository interoperation" work of IMS and OKI. On a more immediate and short-term level, the IMS Digital Library Special Interest Group has a subgroup working on standards for the exchange of "resource lists" (structured lists of readings and similar materials) between course management systems, integrated library systems, and other related entities (resource lists containing pointers to digital resources represent one form of integration of digital resources into learning systems).

In order to further progress in this area, the Andrew W. Mellon provided support for an *ad hoc* group of digital librarians, course management system developers, and publishers to meet and discuss some useful next steps to increase the integration of existing digital resources into the working environments of instructors in higher education. The Group (see *Appendix A* for a list of participants), co-chaired by Dale Flecker of Harvard University and Neil McLean of IMS Australia, met face-to-face twice, in August and December, 2003. It spawned two working groups, each of which wrote a report, as discussed later. This paper summarizes the work of the Group as of March, 2004

General findings and observations

.
Given the breadth of the topic, it is unsurprising that discussions of the Group ranged over a large number of issues. Among the issues and observations that most affected the direction of the Group are:

- The barriers to finding and re-using extant digital materials in a course context are very high today. Locating an appropriate place to look for materials, finding individual resources within the systems in which they are described, reusing existing descriptive metadata, coping with access management systems, understanding technical formats and intellectual property constraints, and ensuring continuing access to selected objects are all difficult. We do not have systems in place which make it easy for instructors with limited time and very limited technical expertise to simply locate and reuse digital content.

- The universe of systems containing materials useful in teaching and learning is highly diverse. This diversity is a reflection of many factors: differing types of digital objects (geographic databases versus art images), different organizations (Harvard's collections versus MIT's), different technical sophistication (an e-journal published by Elsevier versus one published in an academic department), and different intent (legal information systems versus genomic databases). The combination of these factors represents a major challenge in creating a coherent and easily useable information environment for instructors assembling resources for a course.

- The universe of systems containing materials useful in teaching and learning is very large. It includes not just systems internal to universities (institutional repositories, digital libraries, learning object repositories, museums) but also a very large number of commercial and non-commercial publishers which are normally accessed through the library under licensing agreements.

- The diversity of players in the digital domain is an impediment to the wide-spread implementation of any proposed solutions to simplifying the environment. While there are important content holders that are part of the educational environment and who see the support of education as one of their important roles, there are many more players for whom educational use is a minor or incidental part of their business. As we identify measures that content players can take to simplify access and reuse, it will be important for the educational community to work with content holders to explain the importance of, and where possible, induce them to adopt, such measures.

- Simplifying the use of resources goes well beyond questions of repository access protocols and standards. The need to work with system operators and data owners was noted above. The scale of available resources will require considerable management effort in terms of selection, storage and organization. Intellectual property concerns will require efforts in licensing, vendor relations, and education. Heterogeneous systems interfaces and metadata practices will require efforts in data conversion, in the building of agents to cope with diverse interfaces, and in constant monitoring to cope with incessant change. These efforts will necessarily involve many parts of the university.

- Much of the thinking to date in this domain has concentrated on formal course management systems. But we know that instructors use many different kinds of systems to deliver teaching materials, including, for example, the use of PowerPoint for classroom presentation, or of a course weblog to share work.

- Tools and systems relevant to discovering and using information resources are being acquired or created elsewhere in the university, and their integration into the learning systems environment will take effort. Metasearch engines, capable of simultaneously searching multiple systems and of homogenizing retrieved metadata, are potentially key components of integration. OpenURL linking servers, capable of resolving metadata links to on-line or physical resources, are also logical parts of the integrating environment. It is worth noting the growing number of both free and commercial sources (e. g., Serial Solutions, SFX Knowledgebase, TD-Net, Jake) which maintain systematic information about electronic resources including up-to-date interface requirements. These are becoming important parts of digital library infrastructures, and could play an important role in integrating information resources into course environments, as they help deal with the diversity and continual change in the information landscape.

During the first deliberations of the Group, David Greenbaum of the Interactive University project at Berkeley introduced a diagram (see *Figure 1*) which captures many of the points above and helped focus the thinking of the Group.

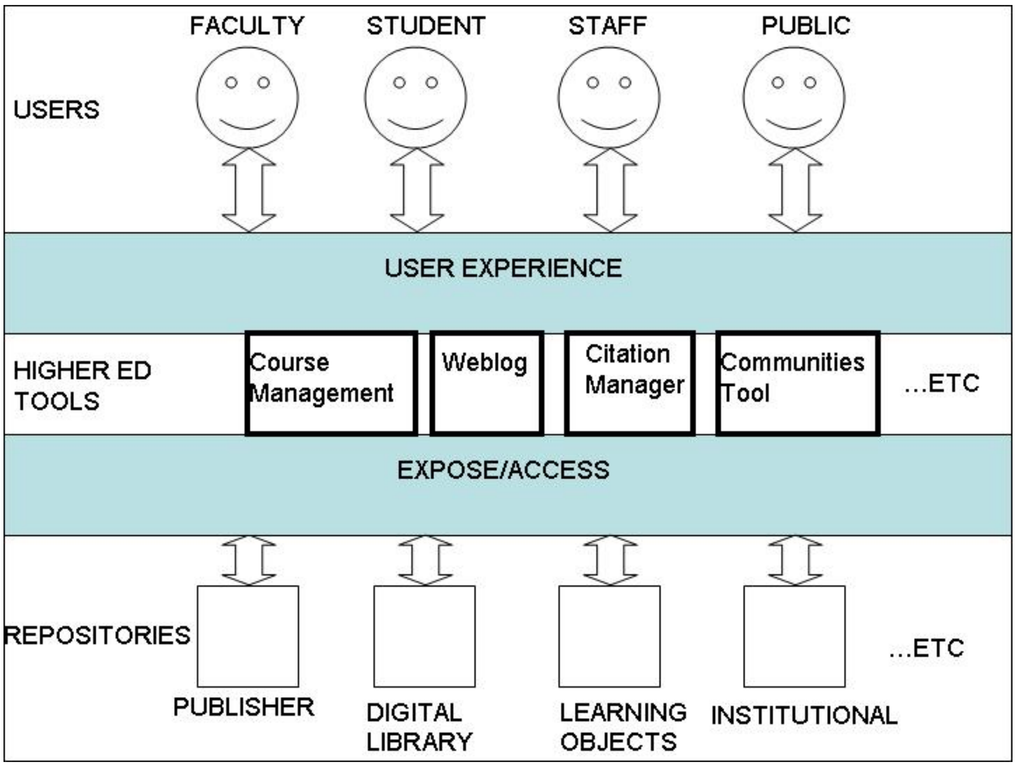

Figure 1. **Users, Tools, and Repositories**

The diagram posits that diversity exists at three levels in the domain: diverse users with different expectations and needs, different tools for users that help meet those needs, and different repositories of useful digital content on which users can draw, frequently with the intermediation of the tools. With this quite satisfying model in front of us, the Group formed two working groups to address the two interface layers in the model: one to think about the interface between repositories and tools, with an emphasis on what repositories should do to make their content optimally useful in such an environment; and one to explore the user experience through the medium of use cases. The latter effort led to an important observation about tools. The work of the two groups is discussed in the next two sections.

Case studies and the need for aggregation tools

In writing the use cases, this work group evolved a model of how resources are gathered and used in teaching. This model is a good deal more complex than the simple "find and incorporate" that is frequently assumed in much of the literature. It is based on three key observations: that relevant digital resources will be distributed over many systems; that the process of using digital objects in teaching usually involves such tasks as arranging, editing, annotating, and describing; and that the results of this work may be used in multiple environments and/or saved for later reuse. The Group defined a general model workflow:

Gather:

DISCOVER = identify content sources
SEARCH = find content within sources
COLLECT = bookmark/link within each content source or within tool, probably using set formats or templates for types of learning objects or aggregations of content
IMPORT = into tool or managed environment, bring or point to content itself, or metadata about content
SAVE = prior to publishing, make a copy for the desktop, external or non-personal workspace that is managed for collaboration or sharing
FIND SIMILAR = identify like items, per the Amazon.com model

Create:

DESCRIBE RESOURCE = annotate, interpret, and write about content before publishing
ORGANIZE = order, sequence, transform content to create learning object
ASSOCIATE = declare link between content or learning object and course, project group or learning objective
MODIFY = change, edit, annotate content or learning object for re-use after initial publishing. Differentiated from **Organize** in that this function may trigger other services to selected community members such as Alerts or Notification related to allowable permissions or conditions to re-use

Share:

EXPORT = transfer content to other formats and/or tools, e.g., PowerPoint, METS. Differs from **Save** by its facilitation of supported format, output, packaging of content or learning object for specific display, rendering, use, storage environments
PUBLISH = make formally available to learning environment with implications for declaration / agreements related to rights for re-use, short and long term storage and archiving services, and expectations for content transformation services
ARCHIVE = establish agreements regarding short or long term storage, preservation, and delivery services.

The Working Group created three specific use cases of the use of digital library content for teaching (the full report of the Group is in *Appendix 4*). Taken together the three use cases illustrate this model and some of its ramifications. The first case describes the process of gathering readings for a humanities course. It involves searching in multiple sources (a library catalog, a digital library repository, and abstracting/indexing databases), and the integration of resources from multiple repositories (including the use of physical as well as digital resources). The second describes the use of a tool to support an instructor's work in aggregating images for use in a class. The tool provides an interface for searching for materials, supports the local aggregation of chosen materials, along with functions of arrangement and annotation, and provides options to output the aggregations in a variety of formats for different purposes. The third use case represents the use of a tool collecting resources that is embedded within a course management system, but that provides the means to search digital content repositories of many kinds including the subscription services managed by libraries.

The work involved in identifying where to look for resources, dealing with multiple system interfaces and varied search functionality, and incorporating heterogeneous metadata and objects into a local environment is enormously complex and rather daunting. The utility of a tool to simplify those tasks is obvious. Many sources, many interfaces, many digital formats are a given in our rich digital environment. Mitigating that complexity and diversity will certainly encourage and enable more instructors to make use of existing resources in their teaching.

A development related to the idea of an aggregation tool for instructors is the growing use of "metasearch" engines in libraries. These engines allow the simultaneous searching of multiple sources, with the engine masking the variations of interface and indexing across the various target systems, and homogenizing the metadata returned as a result of a search. This ability to mask the heterogeneity of many distributed information systems is an obvious part of any aggregation tool. It is worth noting as well the role "knowledge-bases" play in the metasearch environment: databases that contain information needed by the engines to find and use a variety of target systems. There is a good deal of work required to implement metasearch engines, work that will be common as well to aggregation tools. Such tools need to be configured to use appropriate target systems (i.e., configured for local needs and business arrangements), and the knowledge-bases need to be kept up-to-date as target systems change and evolve. An obvious question is how the work related to these distinct but related applications can be combined.

Predecessors of the aggregation systems discussed here are the citation-manager products such as EndNote and ProCite used by many scholars, which support searching, aggregation, homogenization, and flexible output of metadata from a rich variety of sources. Examples of aggregation tools proper are now being created, most notably the Scholar's Box system described in Use Case #2. We expect more such systems to be created in the near term. The Scholar's Box is a stand-alone system. It is easy to imagine similar tools being incorporated directly into course management systems, so that instructors have a unified environment in which to assemble all of the tools and resources needed to support a course.

Considerations for Repositories

As noted above, the universe of digital resources relevant to education is large, growing, and highly diverse. From the vantage point of a developer or operator of a course management system, enabling the use of such resources in local systems will inevitably be daunting. As the Group discussed the challenge and complexity of such integration, it became clear that there were a number of steps that the operators of repositories of content could take that would reduce the difficulties of locating and reusing their content. A work group was formed to analyze in detail what services and practices repository owners should consider when designing their offerings, and to create a checklist for repositories that includes specific standards or best practice recommendations when appropriate. The full report of the Group, including the checklist, is in *Appendix 2*, and a summary version of the checklist has been prepared by Kerry Blinco (IMS Australia).

Figure 2 below shows the relationship of systems and digital objects that the work group addresses. An important element of this diagram is the role of what the work group called gateway systems. These are systems that provide aggregation and discovery services for objects in distributed repositories. Examples of such gateways are abstracting and indexing databases such as Pubmed or Inspec, union catalogs such as OCLC or Melvyl, directories such as MERLOT, and even such search services such as Google. The checklist functions apply as much to these systems as to digital object repositories.

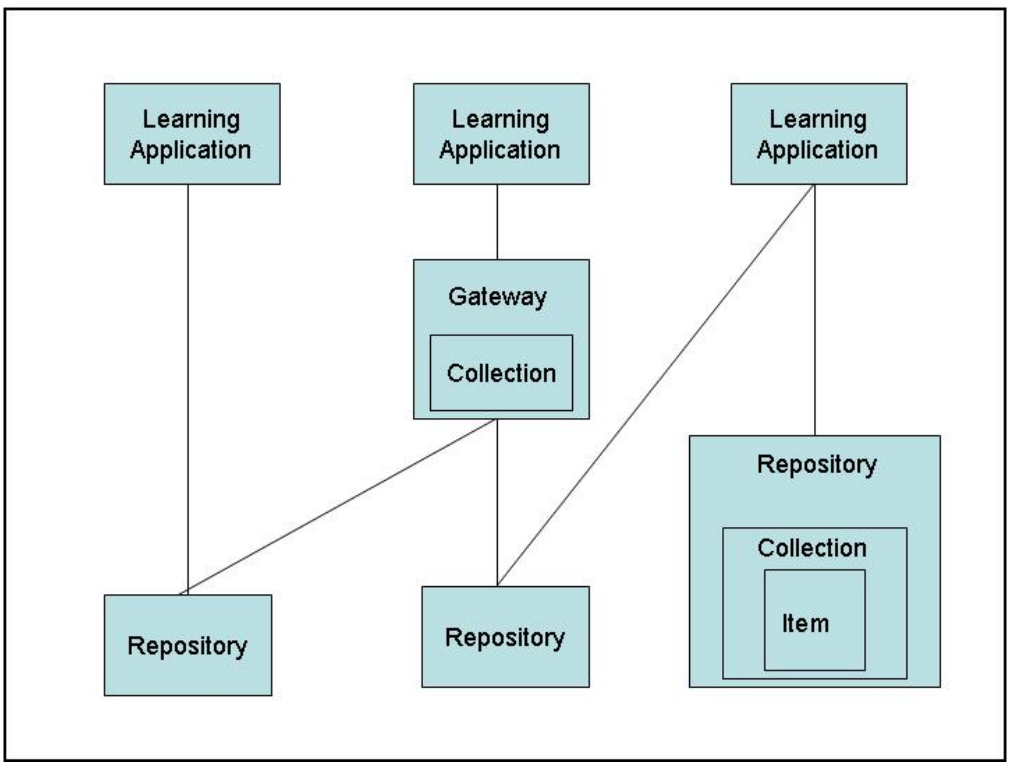

Figure 2. **Learning applications, Gateways, and Repositories**

The work group identified four key types of services relevant to the discovery and reuse of digital resources:

Finding Content. Issues include:

- what types of searching and browsing are supported, and whether complete inventories of content areas are available
- whether standard or well-documented descriptive, structural, technical, and administrative metadata are supported
- whether metadata can be exported for reuse in learning systems
- whether standard or well-documented protocols are supported for searching
- whether metadata is available for harvesting and inclusion in external discovery systems.

Collecting Content. Issues include:

- creation of stable identifiers for content, allowing it to be referenced unambiguously by outside systems
- creation of persistent resolvable identifiers for the location of content which will continue to reference content independent of changes in repositories
- support for standard citation formats, and the export of citations for exchange and reuse.

Accessing Content. Issues include:

- transfer of digital materials to local environments for manipulation and display
- ability of users to specify the format of materials or to download subsets of larger objects
- applications for viewing, utilizing and repurposing objects.

Documentation. Issues include:

- documentation of such critical policies as rights and use, privacy, and security
- documentation of all metadata conventions (vocabularies, subject classification, etc.)
- description of repositories in relevant registries, directories and gateways, so that users know of their existence and what content domains they cover
- description in relevant registries and directories of the technical and policy profiles of the repository (protocols supported, metadata standards supported, access policies) so that applications can interoperate with repositories appropriately.

In addition, the Group identified two general areas of design important for interoperation:

Accessibility. Issues include:

- o does interaction with the repository require the use of proprietary protocols?
- o does the user interface meet recognized accessibility guidelines and legislation?
- o does the repository support standard character encodings?

Access Control. Issues include:

- o does the repository interface to and support standard or conventional external authentication mechanisms?
- o does the repository implement granular authorization rules?
- o does the repository interface to standard or conventional external authorization mechanisms?
- o does the repository interface support anonymous interaction for discovery?
- o is authentication delayed until the point of need?

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, should follow standards and best practices in terms of access, search, metadata practices, and download support, and should 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.

In order to both test the checklist itself, and to get some feel for how the current environment of information systems relates to these criteria, a number of existing repositories in the digital library environment were asked to measure themselves against the list. *Appendix 3* includes the responses from six repositories: ARTStor, the California Digital Library, D-Space, Fedora, Harvard University, and JSTOR. These responses cannot simply be taken at face value for several reasons: they reveal different interpretations of some criteria; some of the responses are for software platforms and many of the criteria are specific to an implementation and some implementations of the software might comply and others; and the interpretations of functions that are "planned" obviously also varies noticeably. Nonetheless, there are a number of interesting observations one can make from the collective responses:

- Even among this population of players, who are likely to be aware of and sensitive to learning applications, no repository comes close to satisfying all of the listed criteria.

- These systems and services are evolving rapidly, and there are plans to implement a number of the criteria that not currently supported.

- One senses that there are a number of areas where, if there were accepted standards or practices that would enhance the usefulness of the repositories for learning applications, developers would be willing to add features to repository applications and that repository managers would be very willing to enhance their services.

- There are some obvious areas where the development of supporting infrastructure and community best practice is required. Preferred metadata formats and content (particularly for administrative data such as rights, and structural metadata for complex objects), a shared understanding of persistent identifiers, registries where services providing digital learning resources can make themselves known and record technical and policy profiles, and the role of and support for software agents to help deal with the large and complex information environment stand out as areas for development.

- Responses from the two institutions with large digital library environments (the California Digital Library and Harvard University) show that such environments are not homogeneous. Both responses repeatedly responded "in some cases", reflecting the variations across their systems. The landscape is complex even within single institutions.

Overall, these responses demonstrate the need for greater awareness of the issues of integration with learning environments, and for more active engagement between the digital library and course management communities.

These the guidelines for repositories and the aggregation tools discussed above are closely related issues. By supporting standards and community conventions and best practices, repositories and related systems can significantly simplify the task of building and maintaining aggregation tools that work across a large environment. The more target systems support standard services and document themselves, the larger the number of targets will be that can be practically supported by such tools.

The need for demonstration projects

The need for improved interoperation between learning systems and digital library systems has been much discussed, but we have today few working examples of such cooperation. As long as these discussions remain theoretical, neither the developers of instructional support systems nor the developers of digital library systems are likely to spend the resources required to support interoperation. We are at a point where some convincing demonstration projects are badly needed. The purpose of such projects include:

- demonstrating the utility of interoperation in the real world. The best argument for supporting interoperation will be instructors who use and care about the functions.

- testing the hypotheses about what functions matter. We need real world experience to see what is actually needed by instructors.

- providing experience with modes of interoperation. While the growing experience with metasearching is beginning to reveal what works and what doesn't in this sort of interoperation, and where we will need additional conventions, standards, and business models, the sort of aggregation tools imagined here will involve many issues beyond simple search, and only experience will show what is needed to support richer interoperation.

- providing a basis for projecting the resources required to implement and support wider interoperation. All the players in this environment are busy and stretched for resources. Interoperation represents a potentially large drain on resources, and systems designers and operators need to understand both the benefits **and** the costs of supporting it.

Our Group is convinced that there is now an adequate base of installed course management systems and of repositories of important educational content to mount meaningful demonstration projects in this domain, and we **strongly** encourage the Mellon Foundation to consider an initiative in this area. We believe that a variety of projects, involving different course management systems as well as a variety of content repositories, are needed. Content sources should include both commercial services (e-journal and e-book suppliers, art image collections, etc.) and university-based systems (institutional repositories, digital libraries, etc.).

Pages 3 – 7 of *Appendix 4* provide a framework created by the Use Case Working Group for considering an appropriate range of demonstration projects.

Other next steps

Discussions touched on a number of other efforts we believe will help further progress in this domain:

- *Use of digital library repositories to support reusable course content*. This Group looked at the interoperation of digital libraries and course management systems from one perspective: the inclusion of content from digital libraries in course environments. There is another potential area of interoperation that has been discussed repeatedly: the storing of materials created in a course context in digital library infrastructures for subsequent discovery and reuse. Such interoperation will involve an entirely different set of issues than those we considered, and we believe that a parallel effort to explore these issues would be beneficial.

- *Communication across domains and stakeholders*. We are struck by how few opportunities there are for digital library and course management developers and commercial information providers to talk systematically about areas of intersection. We found that the various communities did not have a shared understanding of the larger environment, and that we had a great deal to learn from each other's world views. It is not easy to identify how to hold such larger discussions, but we believe an effort in this direction would provide significant pay-back.

- *The need for proselytizing*. Many information providers have little or no understanding of the role of course management systems, nor any appreciation of why making their content easily discoverable and reusable in a course context might matter. There is a need for librarians and course system operators to reach out particularly to the commercial information providers to begin to educate them about the growing role of such systems in higher education.

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.
<u>COLLECTING CONTENT</u>			
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/

Appendix 1: Participants

Lynn Connaway
Consulting Research Scientist III
OCLC

James Fern
Director, OnCourse
Indiana University

Dale Flecker (co-chair)
Associate Director for Planning and Systems
Harvard University Library

David Greenbaum
Director
The Interactive University Project
University of California, Berkeley

Nancy Hoebelheinrich
Metadata Unit Coordinator
Stanford University Library

Leslie Johnston
Director, Digital Services
University of Virginia Library

Amy Kirchhoff
Project Manager
Ithaka Harbors, Inc.

Cliff Lynch
Executive Director
Coalition for Networked Information

Neil McLean (co-chair)
Director
IMS Australia

Jeff Merriman
Project Leader
Open Knowledge Initiative
MIT

David Millman
Director
Information Services R & D
Columbia University

John Ockerbloom
Digital Library Architect
University of Pennsylvania Library

MacKenzie Smith
Associate Director for Technology
MIT Library

Scott Thorne
Lead Architect
Open Knowledge Initiative
MIT

Gordon Tibbitts
President, CEO
Blackwell Science, Inc

Raymond Yee
Technology Architect
The Interactive University Project
University of California, Berkeley

William Ying
Chief Technology Officer
ARTStor

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.

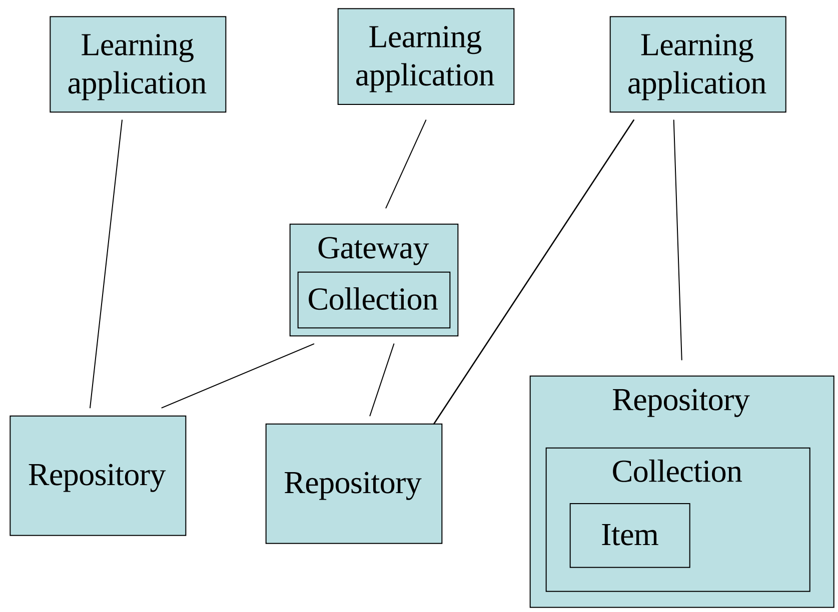

Figure 1: Learning applications search for, collect, and import items contained in collections managed by repositories. Applications can also use gateways that mediate access to virtual collections of items from multiple repositories.

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:

<u>6. Provide ways to access and use content (Essential feature)</u>

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.

<u>Essential functionality</u>: 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:

<u>9. Document policies and functions of the repository (Essential feature)</u>

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.

<u>**Essential functionality:**</u> 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/

Appendix 3

Publishers Note:

This appendix was removed before the document was published on the Digital Library Federation Website http://www.diglib.org/pubs/cmsdl0407/.

instructional media + magic, inc.
25 November 2004

Appendix 4.0: Use Case Working Group: Report and Recommendations

Nancy Hoebelheinrich, Stanford University
David Greenbaum, UC Berkeley
Jay Fern, Indiana University
February 23, 2004

Problem Space Defined:

Faculty, students, and academic support staff increasingly use a broad range of educational technologies to create and share teaching and learning materials. From all reports, however, there is need for new tools and services that will make it much easier for scholars to access and incorporate the rich digital content in digital libraries, museums, and other repositories into their teaching and learning products and practice. While some tools exist to facilitate federated searching, and a number of course management tools help a scholar build and maintain course web sites, scalable tools have yet to be developed which can make transparent the process of gathering and adapting content from digital repositories that is critical for the creation of learning objects.

As scholars become more comfortable incorporating digital content into their teaching and learning products, they are becoming more interested in gathering content from all sorts of providers including those subscribed to by libraries, learning object repositories, academic departmental repositories, their own desktop collections and those sent to them by colleagues in their fields. Once identified, however, there are few common working spaces where scholars can collect the content, transform it as they wish, and create learning objects. At this point, most scholars adapt the content they find in their own working environment using common desktop and presentation tools. Yet, academic support technologists are hearing that scholars want more.

Scholars want to be able to organize, describe, and aggregate content objects into learning objects such as resource lists, case studies, tutorials and lectures. They want to develop subject based themes for the content that they gather as they find it so that it can be used in a number of different venues, from a course to their own research planning. They want to make the discrete and aggregated content in its transformed form available to others in some collaborative, but controlled fashion. Providing tools to facilitate this aspect of scholars' workflow would free their creative processes and, presumably, be of great value to them.

Tools that ease the creation and management of learning content throughout its lifecycle would not only improve the teaching life of scholars, but would also help to define some important services that could be provided by content repositories, and/or by the educational support industry, broadly speaking. Such tools could also help to streamline the conduits by which content and content aggregates are imported and exported into learning environments. By standardizing the packaging of the work products for ease of use in the various environments for which they are intended, the likelihood that content within digital repositories will actually be used by scholars greatly increases, thus bringing better return to the investments already made. In so doing, the chances are much greater that learning objects thus created will be shared by the larger educational community, including K - 12 teachers, community colleges, and international communities.

The problem space being described resides between content repositories on the one side, and educational technologies, of which course management systems are a central component, on the other. The types of content repositories accessed by faculty and students include institutional digital repositories such as that of the California Digital Library and Stanford Digital Repository; learning object repositories such as MERLOT and GEM; library ILS catalogs; text and image content providers such as ARTStor, JSTOR, Highwire Press; scholarly publishers; abstract & indexing databases; departmental repositories of course related materials; personal sources; web pages, etc. On the other side of the space are learning environments and tools in which content has been transformed into learning objects and needs to be associated with learning groups and managed within the learning environment. The learning environment could run the range from a broad Sakai-like Learning Management System to a Course Management system such as WebCT or Blackboard. Other tools for adapting and presenting digital content that faculty and students now use---and would like to use more easily---include such applications as Powerpoint, Endnote, Adobe Acrobat, and weblogs.

Between content repositories and educational technologies lies the scholar's workspace wherein he or she needs to be able to gather content with as much relevant metadata as possible, use and adapt it as necessary (and supportable), interoperate with other tools and environments, and publish it to the desired audiences. It is in this space that the Mellon LMS – DR Interoperability Study Group recommends some demonstration projects be funded. To support those findings, the Study Group has created a checklist of services that a content repository could and should provide to work best with a learning management system and other educational technologies. To complement and ground the analysis of services into real life situations, three initial Use Case Scenarios have been developed from the point of view of scholars who are trying to develop a learning resource. Besides describing the players and transactions depicted by the Use Case descriptions, the Use Cases delineate the functional requirements for a *gathering / authoring tool* that could be developed for learning content creators. (See Use Case Example – 01 through 03)

Functional Requirements and Key Common Assumptions: The following functional requirements emerged from the Use Case Scenarios developed by the group:

Gather:

DISCOVER = identify content sources
SEARCH = find content within sources
COLLECT = bookmark/link within each content source or within tool, probably using set formats or templates for types of learning objects or aggregations of content
IMPORT = into tool or managed environment, bring or point to content itself, or metadata about content .
SAVE = prior to publishing, make a copy for the desktop, external or non-personal workspace that is managed for collaboration or sharing
FIND SIMILAR = identify like items, per the Amazon.com model

Create:

DESCRIBE RESOURCE = annotate, interpret, and write about content before publishing
ORGANIZE = order, sequence, transform content to create learning object
ASSOCIATE = declare link between content or learning object and course, project group or learning objective
MODIFY = change, edit, annotate content or learning object for re-use after initial publishing. Differentiated from **Organize** in that this function may trigger other services to selected community members such as Alerts or Notification related to allowable permissions or conditions to re-use

Share:

EXPORT = transfer content to other formats and/or tools, e.g., PowerPoint, METS. Differs from **Save** by its facilitation of supported format, output, packaging of content or learning object for specific display, rendering, use, storage environments
PUBLISH = make formally available to learning environment with implications for declaration / agreements related to rights for re-use, short and long term storage and archiving services, and expectations for content transformation services
ARCHIVE = establish agreements regarding short or long term storage, preservation, and delivery services.

These functions point to a type of tool (and/or service) that at its core level would enable faculty, students, and the public to *Gather-Create-Share:* to gather a wide range of cultural and scientific digital objects from many different repositories to create teaching and learning products that can be shared with, and reused by, others inside and, in important cases, outside of the higher education community.

The Use Case Scenarios also include underlying strategic assumptions related to the creation of a gathering tool, from the points of view of tool developers, end users, and content repositories. These assumptions were included to better explicate which player could /would be responsible for which function.

Demonstration projects: Recognizing that there are many ways a gathering / authoring tool could be built, it makes great sense to get together a number of collaborative partners not only to ensure that efforts are not duplicated, but also to make sure that the tool can work together with other major efforts in the digital repository / LMS arenas. Some early and promising prototypes exist, such as the UC Berkeley Interactive University's *Scholar's Box* tool or Groxis' *Grokker* product, but these tools either need further development to provide scalable options for scholars (Scholar's Box), or do not yet have the hooks into the Learning Management side of the equation to meet scholars' needs (Grokker). It would make sense to evaluate what has been created already to assess what functionalities are included and what are not, how well the features provide the services and then determine what the focus should be for various partners contributing to the demonstration projects. Included with this summary are a number of brief use case scenarios that have been sketched to illustrate the functionalities noted above.

Collaborative partners: Many useful combinations of collaborators could be envisioned to achieve the functionalities described by the Use Case Scenarios, and indeed, discussions among some of the institutions seated at the Study Group table have already begun. Some of the more obvious partners include content provider repositories of various formats and degrees of preservation responsibility, integrated library systems with course reserve and catalog modules, federated searching tools, portal tools, course and / or learning management system tools, and of course, scholars from various disciplines who are interested in a variety of content formats.

Lifecycle Work Flow: An approach that could guide demonstration projects is to explore the lifecycle or end-to-end process of higher education faculty, students, and/or staff gathering materials from multiple repositories, creating new teaching and learning products from these materials (in some cases through use of other authoring tools as well), and then sharing/publishing materials for reuse in some sort of repository, whether it is part of a digital library, learning object repository, and/or a repository within a course management system. In each case, these end-to-end demonstrations (from repository-to learning environment-to repository) would be driven by the needs of key users (use cases).

Demonstration projects would then include the following major components in the end-to-end process, with choices to be made in each area about what to focus on:

Use Cases (user experience) as driver →
→ Material/Object Type →
→ Sources/Repository technologies →
→ Instantiations of "Collect-Author-Share" tool →
→ Interoperation with other educational and information technologies →
→ Publishing to Learning Material/Learning Object Repositories
→ {This cycle can then repeat in various re-use scenarios}

Demonstration projects do not have to follow this complete functional flow; one might, instead, focus on some of the steps in this process. We believe there is substantial value, however, in exploring, from both the user's point of view and the technologist's point of view, the complete gathering-creating-sharing cycle(s) within which flexible teaching and learning materials are built from the rich digital content collected in digital repositories and museums.

Below is a list of possible options for each major component in the lifecycle process. By choosing from these options (a menu of sorts), different demonstration projects could be crafted, comprising a range of partnerships that explore important use cases and instantiations of technology. This is not meant to be a comprehensive list (nor is it in priority order), but rather includes some interesting choices from the working group's point of view.

Menu to Choose from for Demonstration Projects

1. **Use cases/User Experiences** (that would drive/be investigated in pilot(s))

    a. Faculty and graduate students create lecture materials from collections of digital images and associate with syllabus inside campus CMS managed course web site
    b. Faculty and/or graduate students create reading and/or resource lists from within a Library portal and publish to an independent course web site
    c. Faculty browse interdisciplinary Reading List collections within learning object repository (e.g., OCW at MIT) and select one for reuse in a collaborative research project among scholars on different campuses
    d. Faculty creates a pre-defined search from various content repositories using metasearch tool and adds to an assignment in the CMS managed course web site
    e. Students gather images, text, citations from multiple sources to develop a joint presentation for class
    f. Library curator creates a themed collection of learning materials based on her subject specialty and publishes it to Library's Subject Resources web page for use by wide range of scholars
    g. Instructors from a community college build small learning objects from materials in the themed collections found on a Library's publicly accessible web page and publish them to a learning object repository (e.g. MERLOT)
    h. Students gather class presentation materials for use in e-portfolio tools

2. **Content types** (both public and licensed materials)

    a. Images
    b. Bibliographic resources
    c. Text – journal articles, books, other
    d. Audio
    e. Video

f. Pre-defined searches into content repositories, possibly using metasearch tools
g. IMS Content packs and other complex structured objects
h. METS objects

3. **Content Repositories** (that could provide representative content, explore important repository technologies or services, and provide connections to important educational technologies)

    a. UC California Digital Library (brings in multiple UC campuses)
    b. Stanford Digital Repository
    c. Indiana University's Library Collections
    d. Harvard University's Visual Resources collections
    e. DSpace (MIT, Columbia, and other institutions; and, in future, OCW materials from MIT)
    f. University of Virginia representing Fedora
    g. ARTStor
    h. JSTOR
    i. Luna collections
    j. Open Archives Initiatives (OAI) repositories
    k. Learning Object repositories: MERLOT, EdNA (Australia), EduSourse (Canada), SMETE
    l. MusicSTOR (Michigan State)
    m. Public Library of science and other open repositories
    n. NSDL collections
    o. Social Science dataset collections
    p. Unstructured web materials (while still trying to preserve context)
    q. Individual desktops / personal collections
    r. Johns Hopkins Project MUSE (humanities)
    s. Publishers, e.g., Blackwell Publishing

4. **Possible Models for Instantiating Gather-Create-Share Tools/Services** (these might take the form either of web or client based tools/services.)

    a. Integrated in Course Management Systems
    b. Portlet / tool in Sakai portal framework
    c. Component and/or closely coupled with digital library (able to talk to other digital libraries as well)
    d. Stand alone tool able to interoperate with educational technologies and digital libraries

5. **Authoring Tools with which to Interoperate/Integrate**

    a. Powerpoint
    b. EndNote
    c. Adobe Acrobat/PDF
    d. Weblogging tools/RSS

e. Reload – IMS Learning Design based authoring tool
f. Chandler/Westwood
g. VUE (Tufts)
h. TK3 (The Night Kitchen)
i. OpenOffice.org
j. Meta-search tools
k. UCB Scholar's Box tool (which is both an early version of a *Gather-Create-Share* tool and an authoring tool other *Gathering* tools might interoperate with)

6. **Other Integrative Environments with which to Interoperate**

a. Sakai (which then includes the CHEF / OnCourse /CourseWork CMS framework, research groupware, UPortal, OKI/ IMS Abstract Framework)
b. Chandler (as a platform)
c. LionShare peer to peer network infrastructure
d. E-Portfolio (Indiana and others)
e. Library Integrated Library Systems (including catalog, electronic reserves)
f. Library and other departmental portal environments
g. Semantic Web – for example, SIMILE project and their Haystack tools at HP/MIT)

7. **Repositories in which to Publish / Share Teaching and Learning Materials**

a. Institutional repositories
b. Campus departmental repositories
c. Learning Object repositories (e.g., MERLOT, GEM)
d. Course and learning management system repositories

Appendix 4.1: Use Case Example #1 -- Adding an Online Resource List to a Humanities Seminar Course Website from within a University Library's Portal

Nancy Hoebelheinrich, Stanford University
February 22, 2003

Players:

- Faculty member (Professor X) who created bibliography on paper for Humanities Seminar she is teaching.
- Graduate Teaching Assistant who is supporting both the faculty member in teaching the class and the students in the class with their research.
- Students in the Humanities Seminar who must access and read the resources on the Resource List
- University's Integrated Library System (ILS) Online Catalog
- University Libraries' Metasearch application
- University Libraries' Digital Repository
- University Libraries' Course Reserve Lists
- University Libraries' Subject Resource Lists
- Campus CMS

Assumptions:

<u>Tool Developer's Point of View:</u>

- Hooks exist within the Libraries' metasearch application to allow two way interaction between it, and the CMS.
- The Libraries' metasearch environment includes:
    - the Library catalog for physical and digital versions of books, locations for print versions of journals
    - various sources for electronic journals
    - abstract and indexing databases
    - digital repository content
    - Library Course Reserve Lists
    - Library subject based resource lists
- Within the CMS, another tool or module is provided which creates and manages Resource Lists and links a given RL to course web site(s) within the CMS.
- Search results from the Libraries' metasearch application are returned to the CMS in an agreed upon, supported set of display formats including citation, HTML pages, etc.

- An agreed upon bibliographic citation format is supported for display of content metadata in the CMS etc.
- A URL for the content during the term can be constructed from the content metadata or imported into the CMS, either as a link or as content per se.
- Content delivery transformation services are built into the CMS from likely content providers delivered in supported packaging protocols.
- Appropriate course, registration, authentication, authorization, and licensing information is automatically shared among campus systems.
- Means to "publish" a Resource List (as a learning object) within the CMS or outside of it provided as a service of the CMS.
- The scope and requirements for publishing a Resource List is defined in conjunction with other campus institutions including the Libraries' Digital Repository, the campus scholarly community, etc.
    - Range of publishing scope includes
        - Publishing within CMS environment
        - Publishing to outside of CMS environment to personal, e-portfolio or scholarly publishing type of environment
        - Publishing to outside of CMS environment in to long term preservation storage

<u>End User (Professor / TA/Student) Point of View:</u>

- Professor X created and used the bibliography last term at which point all the texts were available either through Library Course Reserves or in the CoursePack, as indicated on the paper bibliography.
- The TA has been proxied to create the Resource List in the campus CMS for Professor X.
- All of the authentication processes related to creating the Resource List come transparently from the campus infrastructure.
- The TA has significant experience conducting online and library catalog research.
- Students will need access to the texts and or metadata on the Resource List for the entire term.
- Searching the library's resources appears the same and has the same search options whether search is done w/in RL creation tool, the Library catalog, the Libraries' metasearch tool or any of the Libraries' Course Reserve or Subject Resource Lists.
- Professor X plans to re-use the Resource List next time the Humanities Seminar is offered the following year.

<u>Content Repository Point of View:</u>

- The Content Repository assumes that there are multiple types of users - from the general public to university faculty and staff, to software agents such as Metasearch tools.
- Content Repository supports industry and service-level standards for resource / item level descriptive metadata.

- The Content Repository exposes its resources by providing a core set of search/browse and delivery services to its customers / patrons.
- Extent and scope of the content search and delivery services to the user are defined by user's identify and authorization, and managed by commonly used security and authentication standards.
- Resources made available by the Content Repository for inclusion in a learning management system are packaged in standard formats for dissemination to an external system for transformation and delivery to the end user(s).
- The repository assumes that services that connect to it will not interfere with the repositories' operations.

Description:

Professor X created and distributed a paper based bibliography of journal articles, entire books, chapters from books, and selected poems last term for her Humanities Seminar that she wants to make available through the online course web site managed by the campus course management system that she is using this term for the Seminar. She has asked her seminar TA to turn the bibliography into a Resource list that will allow the students to find whether and where the print texts are located on campus, provide a proper citation according to the Chicago Manual of Style citation format, and include the campus location. She also wants to add other resources that she has read since last term, but now does not know exactly where the texts of the resources can be found, so does not have complete citation information. If the resources are available in digitized form, Professor X would like that information included in the same citation format along with the link to the online version.

Transactions:

- Because Professor X has included a number of publications from the same authors who are considered specialists in the topic areas on the bibliography, the TA searches the library catalog by author to locate any of the resources on the bibliography from the given author.
- From the results of the library catalog search, the TA selects a number of the books which are located in various campus libraries, downloads the descriptive information about the book and exports that information into the Resource List creation module of the CMS. The TA chooses the citation export format for Chicago Manual of Style, and opts to include the campus library location as part of the export. The TA also chooses the option of adding the selected books to a Library Course Reserve List for the Humanities Seminar that can be automatically generated from the CMS' course information about the Humanities Seminar.
- Because Professor X has excerpted selected chapters from some of the books, the TA uses the Resource List module from the CMS to correct the citation from the whole book that has been exported into the Resource List and limit the citation to the chapters for which the students are responsible. The TA also chooses the option of adding the selected chapters to the Library Course Reserve List for the Humanities

Seminar automatically generated from the CMS' course information about the Humanities Seminar.

- The TA suspects that some of the complete books or chapters may be available from the Library's online repository in digitized form. She conducts the same author search in the Library's digital repository. If she finds a digital version of the text or selected parts of the texts, she uses the DL's "Send a Shortcut" button to export the location of the text, and chooses a "CMS Resource List" as the place where the location should be dropped. Upon being returned to the Resource List module, the TA identifies to which Resource List and citation the location should be added. She is then given the option to return to the Library portal for further searching, or completing the creation / editing of the Resource List within the CMS.
- The TA opts to return to the Library portal and now searches it to find the known journal titles of the articles that Professor X has requested. Once she finds a journal, she searches for the article title by whatever means the journal provides. Once she locates the desired article, she chooses the options for exporting the article's citation information in the desired citation format, and exports that metadata. As part of the export process, the TA is returned to the portal where she chooses the option of dropping the citation information and link to the article into the Resource List module of the CMS. Following the completion of this task as outlined above, she is offered the option of further searching from within the portal or completing the creation / editing of the Resource List.
- The TA returns to the Library's portal to find the articles from the unknown journals that Professor X wants to add to the Resource List this term. She locates the appropriate abstracting and indexing database, and then searches the database for the article. From the results list, she follows the link to the article and upon determining that it is the article desired, chooses the option for exporting the article's citation information in the desired citation format, and exports that metadata. As part of the export process, the TA is returned to the portal where she chooses the option of dropping the citation information and link to the article into the Resource List module of the CMS. Following the completion of this task, she is offered the option of further searching from within the portal or completing the creation / editing of the Resource List.
- As a final step in the creation of the Humanities Seminar Resource List, the TA verifies that the time during which the cited texts are needed by students in the Seminar is the current term or less. If more time is needed, the TA is referred to the Library's licensing specialists for consultation.

Exceptions:

- If the expiration date for access to the resources on the Resource List is later than the end date of the current term, the Resource List is not mounted on the Humanities Seminar's course web site until cleared by the Library's licensing specialists.

Appendix 4.2: Use Case Scenario #2 -- University Faculty Using IU Scholar's Box To Create Reusable Teaching Materials With Images

David Greenbaum, Jane Lee, Chris Ashley
Interactive University Project, UC Berkeley
December 10, 2003

Players and Tools

- Professor Chris Jones (History/Ethnic Studies/Sociologist)
- IU Scholar's Box Tool
- UC California Digital Library (collections from multiple UC campus's and other collections)

Assumptions (Note – we have intentionally included a large number of sometimes broad assumptions to push forward discussion.)

Tool Developer Point of View

1. The developer assumes that the Scholar's Box tool is not part of a digital repository or Course Management System (CMS) per se, rather it bridges/interoperates with them. The SB can be designed to be loosely or closely coupled digital library and/or CMS.
2. The Scholar's Box (SB) can be a client and/or web-based tool. In this use case, the SB is a client based tool that can "save" and "publish" to a learning object repository for reuse by others.
3. The Scholar's Box can interoperate with a number of common productivity, research, and social software applications, e.g., Powerpoint, EndNote, weblogs.
4. The developer assumes that users want a tool that's easy to use and has a clean design, especially in relation to authoring of learning materials. (It is also assumed that more complex authoring of collections and learning objects does not work very well as a web application.)
5. The developer assumes that repositories deliver content/metadata in a number of standard XML formats.
6. The developer assumes that there are multiple types of users - from the general public to university faculty and staff.
7. The developer assumes that individual users will have multiple collections and that the Scholar's Box needs to support a user account structure.

8. The developer assumes that end users will want to share part or all of their collections and teaching and learning objects with others inside and outside their institutions, and that users will want ability to specify what materials can be shared.
9. The developer assumes that personal collections as well as learning objects are important products to be created and shared, and that in each case there is real value in making available these products in XML formats that can be easily disaggregated and re-aggregated.
10. The developer assumes that almost all teachers and learners carry out some form of personal collecting, organizing, annotating, interpreting, presenting, and sharing ("Scholarly Primitives") in their work with images (as well as other object types), and that tools should be developed to make it much easier to do this with digital cultural and scientific objects.

End User (i.e. Professor) Point of View

1. The user assumes that the Scholar's Box can interoperate with local CMS/LMS(s).
2. The user assumes that the Scholar's Box is connected to multiple digital libraries/repositories, such as the California Digital Library.
3. The user assumes that he or she is able to export his or her collection to a learning object repository and publish it, for example, in commonly used and interpreted format, such as an IMS content package.
4. The user assumes that he or she can access all the digital content that he or she normally does in the library.
5. The user assumes that he or she can see that objects are specified as publicly accessible or private (have restricted use).
6. The user assumes that he or she can decide what level of access (permissions) to give to his or her collections and learning objects.
7. The user assumes that he or she can incorporate his or her own material from his or her personal computer.
8. The user assumes that desktop client tools can integrate with the Scholar's Box.

Content Repository Point of View

1. The repository assumes that there are multiple types of users - from the general public to university faculty and staff.
2. Repositories support industry and service-level standards.
3. The repository assumes that services that connect to it will not interfere with the repositories operations.
4. The repository assumes that services that connect to it will maintain security and authentication standards.
5. Some campus repository service kindly agrees to host learning objects and other related course materials so that others can access for them on-gong use/re-use.

Description

Professor Jones wants a collection of images for lecture she is giving on Angel Island and its essential role in early Californian immigration. She enters "angel island immigration" into the search box and chooses “CDL” from the checklist of repositories available through the Scholar’s Box. She could have searched more repositories, but she knew that the CDL provides access to some of the richest and most relevant primary source and other materials related to California’s immigrant history. The search returns a rich set of images and some related textual and bibliographic materials (this use case focuses on images, but it assumed that even users primarily concerned with images will want to be able to also easily gather other related materials/object types).

As Professor Jones gathers images for her Angel Island lecture, she encounters interesting material that she would like to incorporate into other future lectures. She finds herself following threads that aren't necessarily directly related to her immediate teaching task, but interesting nonetheless; she creates other "collections" for these other objects and uses. Professor Jones also notices images that may relate to her current research and drags those images into her preliminary research collection and writes a sentence or two about her initial thoughts on the items in the “Notes” field. She eventually returns to the "collection" containing photos for her lecture and starts to annotate them. After she is finished, she will have the Scholar’s Box automatically create a slideshow presentation using Powerpoint and handouts of the images and her annotations for her upcoming lecture. She will also make available the set of images and notes for her class in the CMS she is using. Finally, she will make available this collection of images for other colleagues at other academic institutions to use and modify.

Transactions

1. Professor Jones opens up the Scholar’s Box (SB) client on her laptop.

2. After reviewing the collections she has already saved in other uses of the SB, Professor Jones creates a new collection by choosing "New Collection" from the File menu.

3. Professor Jones goes to the search box and begins entering search terms. The results list shows thumbnails of the images found as well as some metadata, such as title and source. She drags the thumbnail images that she wants to use from the search results list to her collection window. The image along with all of its metadata is shown in the collection window.

4. After collecting all the images she wants, Professor Jones annotates selected images in the collection by entering the desired text into a text box labeled "Annotations". (Each image has its own "Annotations" box.) Her annotations automatically become part of the metadata for their respective images in her Angel Island collection.

5. Professor Jones saves her collection as "Angel Island Lecture" by choosing "Save" from the file menu.

6. Professor Jones chooses "Create slideshow of collection" from a menu of services that can be applied to collections in the SB to create a slideshow presentation of her Angel Island collection. (The collection of images is exported by the SB into OpenOffice presentation tool and/or MS PowerPoint.)

7. Professor Jones reviews the results and fine tunes the layout of some of the slides by using the appropriate Powerpoint procedures.

8. Professor Jones returns to her collection in the Scholar's Box and chooses "Create handouts" from a menu or clicks "Create Handouts" button. Scholar's Box takes images and information from her collection and automatically creates handouts of the images in the collection.

9. Professor Jones then saves her Angel Island lecture collection to the local CMS she and her students use so that the students can access the full collection of objects for possible other uses via the SB and/or the CMS (note, the SB can save and export the collection as an IMS content pack so that IMS compliant tools/environments can integrate these materials).

10. Professor Jones then saves several of the collections she has created in an external learning/teaching object repository so that she can access these materials from other computers she uses (note, the SB can also save this collection as an METS object so that it can better interoperate with various digital repositories).

11. Professor Jones also decides that she wants to share one of these collections with colleagues of hers at other universities and teaching institutions. She is able in the campus LO/courseware repository to make collections available to the public via the web and/or create and specify groups who can have access to these materials.

Appendix 4.3: Use Case Example #3 – Faculty creates a search criteria box and adds a search criteria box to a CMS assignment

Jay Fern, Indiana University
December 10, 2003

Players

- Faculty
- CMS
- Search Creator Tool
- Digital Repository (DR) and vended Databases.

User (faculty) Assumptions

- The faculty has located the Search Creator Tool in the CMS, authenticated to the enterprise, and is ready to a Search Criteria Box.
- A Search criteria Box that has been published to a specific user or group is editable by a subscribed author.

Tool Developer Assumptions

- Search Criteria boxes are associated with an individual or group.
- The Search Creator tool has the capacity to:
  - Browse/Subscribe - Browse current Search Criteria Boxes that have been associated with groups (a particular role) and subscribe to them.
  - Create/Edit/Publish – Create/edit a Search Criteria Box and publish that list to specific groups.
  - Manage – Select preferences, add search criteria to those that are subscribed.
- The enterprise has an established groups (Authorization) mechanism.
- The SIS (Student Information System) is interoperable to the enterprise.

Content Repository Assumptions

- The repository assumes that the users are authenticated to the enterprise.
- Repositories support industry and service-level standards.
- The repository assumes that services that connect to it will not interfere with the repositories operations.
- The repository assumes that services that connect to it will maintain security and authentication standards.

Description

Professor Thomas wants to create a Search Criteria Box (SCB) for his Biology 101 course to help students focus their research activities around specific journals, repositories and other federated search engines. His goal is to guide his students and help them use their time wisely in creating research assignments His colleague, Professor

Smith, has created a similar SCB for a course last semester and has subsequently published it for his colleagues to adapt at will. Professor Thomas would like to utilize his work yet will be editing it to include additional journals. When he completes the editing, he will place the SCB in the CMS for assignment, week 8.

Transactions

1. Professor Thomas selects the "browse Search Criteria Box" button and is presented with a collection of SCBs and chooses to sort by author.
2. He locates the "Smith; Biology 101" SCB and subscribes to it.
3. Next he selects Manage SCBs which presents a set of SCBs he has created or subscribed to. He locates "Biology 101" and then selects "edit." The edit function allows him to change the search criteria, associate the SCB with groups, set preferences and add resources.
4. He selects the "Define Search Criteria" button and is presented with a focused search string associated to a set of Digital Repositories and Vendor Database journals that Professor Smith has created.
5. Professor Thomas modifies the string to focus his students to cell biology. He saves his changes and is returned to his Search Criteria Box master list.
6. He clicks on "assignment" in the CMS and browses for "Assignment: Week 8."
7. He clicks to edit the assignment and is presented a WYSIWYG editor. He places his cursor after the second paragraph of the assignment text and clicks the "Add Resources" button.
8. He is prompted to choose from a list of resources including a file from his local storage or network drive, a URL, Resource List artifact or Search Criteria Box. He selects the SCB option and clicks OK.
9. Professor Thomas is presented with a list of SCBs to which he is currently subscribed. He scrolls through the list and selects the "Biology 101" SCB and clicks "Select." The assignment text returns to the screen with the Biology 101 SCB centered after the second paragraph.
10. He clicks "Save" and is returned to a preview of the assignment.

Exception: if the user not a registered user (authenticated) in the enterprise, he will not be able to create and publish.