



IMS Digital Repositories Interoperability - Core Functions Best Practice Guide

Version 1.0 Final Specification

IPR and Distribution Notices

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

IMS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on IMS's procedures with respect to rights in IMS specifications can be found at the IMS Intellectual Property Rights web page: http://www.imsglobal.org/ipr/imsipr_policyFinal.pdf.

Copyright © IMS Global Learning Consortium 2006. All Rights Reserved.

If you wish to distribute this document or use this document to implement a product or service, you must complete a valid license registration with IMS and receive an email from IMS granting the license. To register, follow the instructions on the IMS website: <http://www.imsglobal.org/specificationdownload.cfm>.

This document may be copied and furnished to others by Licensee organizations registered on the IMS website provided that the above copyright notice and this paragraph are included on all such copies. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IMS, except as needed for the purpose of developing IMS specifications, under the auspices of a chartered IMS work group.

Use of this specification to develop products or services is governed by the license with IMS found on the IMS website:
<http://www.imsglobal.org/digitalrepositories/driv1p0/driv1p0speclicense.html>.

The limited permissions granted above are perpetual and will not be revoked by IMS or its successors or assigns.

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

**Copyright © 2003 by IMS Global Learning Consortium, Inc.
All Rights Reserved.**

The IMS Logo is a registered trademark of IMS/GLC.

Document Name: IMS Digital Repositories Interoperability - Core Functions Best Practice Guide

Date: 13 January 2003

Table of Contents

1.	INTRODUCTION	3
1.1	NOMENCLATURE.....	3
1.2	REFERENCES.....	4
2.	TECHNOLOGY RECOMMENDATIONS	5
2.1	XQUERY	5
2.2	Z39.50	5
2.3	SOAP.....	6
2.4	OAI MODEL REQUIREMENTS.....	6
3.	EXAMPLES	7
3.1	SOAP MESSAGING FOR XQUERY SEARCH USING IMS META-DATA	7
3.2	SUBMIT/STORE USING FTP WITH IMS CONTENT PACKAGING	8
3.3	Z39.50 BIB-1 ATTRIBUTE SET USE ATTRIBUTES EXTENSION FOR IMS META-DATA ..	8
3.4	CROSS-DOMAIN SEARCH USING IMS META-DATA ACCESS POINTS	11
3.5	LEARNING OBJECTS NETWORK EXAMPLE.....	12
4.	FUTURE CONSIDERATION.....	14
4.1	PROVISIONS AND DISCLAIMERS	14
4.2	REGISTRIES AND DIRECTORIES.....	14
4.2.1	<i>Identify and Locate Services</i>	<i>14</i>
4.2.2	<i>Identify and Locate Objects.....</i>	<i>15</i>
4.2.3	<i>Identify and Locate People.....</i>	<i>15</i>
4.2.4	<i>Identify and Locate Resources</i>	<i>15</i>
4.3	DIGITAL RIGHTS MANAGEMENT	16
4.4	LOCATION AND RESOLUTION SERVICES	16
4.4.1	<i>DOI.....</i>	<i>16</i>
4.4.2	<i>OpenURL.....</i>	<i>17</i>
4.4.3	<i>PURL.....</i>	<i>17</i>
4.5	REQUEST/DELIVER SERVICES.....	17
4.6	WEB SERVICES	18
4.7	RECOMMENDATIONS REGARDING GUID ALLOCATION	18
5.	SUPPORT FOR ROLES.....	20
5.1	LEARNER	20
5.2	CREATOR	20
5.3	INFOSEEKER	20
5.4	AGENT	20
6.	CONFORMANCE ISSUES	21
6.1	NO CONFORMANCE STATEMENT.....	21
6.2	APPLICATION PROFILING	21
6.3	IMPLEMENTATION ISSUES.....	22
	APPENDIX A – GLOSSARY	23
	APPENDIX B – POSSIBLE FUTURE DIRECTIONS	25
	APPENDIX C – ADDITIONAL RESOURCES.....	28
	ABOUT THIS DOCUMENT	29
	LIST OF CONTRIBUTORS	29
	REVISION HISTORY	30
	INDEX.....	31

1. Introduction

This document constitutes the Best Practice and Implementation Guide for the Phase 1 specification of the IMS Digital Repositories Interoperability (DRI) Project Group. Phase 1 focus is on the core functional interactions between the Mediation and Provision layers of the Functional Architecture (see the DRI Information Model). This specification is intended to utilize schemas already defined elsewhere (e.g., IMS Meta-Data and Content Packaging), rather than attempt to introduce any new schema.

Section 2 of this specification offers guidance on XQuery, Z39.50, and SOAP, which are the key technologies identified for implementation of the core functions.

Section 3 identifies technologies, which are not included in the current specification, but are flagged up as under consideration for future phases of the DRI work. It should be emphasized that these are candidate technologies and their inclusion here is not a commitment to their future adoption, but rather an indication of the current forward view of the group.

Section 4 attempts to relate the technologies adopted to the user experience of using these core functions.

Section 5 explores the various roles supported by the specification.

Section 6 discusses the issues of conformance as they relate to implementing the specification.

Appendix A offers a glossary of terms, which are used throughout the specification.

Appendix B identifies some of the issues that could possibly direct the future development of the specification.

Appendix C refers to additional resources that are important to the specification.

1.1 Nomenclature

ANSI	American National Standards Institute
AP	Application Profile
DOI	Digital Object Identifier
DRI	Digital Repositories Interoperability
FTP	File Transfer Protocol
GUID	Globally Unique Identifier
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
LDAP	Lightweight Directory Access Protocol
LOM	IEEE Learning Object Metadata
MIME	Multipurpose Internet Mail Extensions
NISO	National Information Standards Organization
OAI-PMH	Open Archive Initiative Protocol for Metadata Handling
REL	Rights Expression Language
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
W3C	World Wide Web Consortium
XQuery	XML Query

1.2 References

- [DRI, 03a] *IMS Digital Repositories Interoperability - Core Functions Information Model*, K.Riley and M.McKell, Version 1.0, [IMS](#), January 2003.
- [DRI, 03b] *IMS Digital Repositories Interoperability - Core Functions XML Binding*, K.Riley and M.McKell, Version 1.0, [IMS](#), January 2003.
- [MD, 01a] *IMS Learning Resource Meta-Data Information Model*, T.Anderson and M.McKell, Version 1.2.1, [IMS](#), October 2001.
- [MD, 01b] *IMS Learning Resource Meta-Data XML Binding*, T.Anderson and M.McKell, Version 1.2.1, [IMS](#), October 2001.
- [MD, 01c] *IMS Learning Meta-Data Best Practice and Implementation Guide*, T.Anderson and M.McKell, Version 1.2.1, [IMS](#), October 2001.

2. Technology Recommendations

2.1 XQuery

XML is an extremely versatile mark-up language, capable of labeling the information content of diverse data sources including structured and semi-structured documents, relational databases, and object repositories. The W3C XQuery language uses the structure of XML intelligently to express queries across all these kinds of data, whether physically stored in XML or viewed as XML via middleware. Because query languages have traditionally been designed for specific kinds of data, most existing proposals for XML query languages are robust for particular types of data sources but weak for other types. The W3C XQuery Language, is designed to be broadly applicable across all types of XML data sources, including both databases and documents. XQuery is designed to be a small, easily implementable language in which queries are concise and easily understood. XQuery is also based on a strong formal model (see XML Query 1.0 Formal Semantics).

The principal forms of XQuery expressions types, including the following:

- Path expressions, based on the syntax of XPath (see XQuery 1.0 and XPath 2.0 Data Model).
- Element constructors to enable a query to generate new elements.
- FLWR (pronounced “flower”) expressions constructed from FOR, LET, WHERE, and RETURN clauses, which must appear in a specific order.
- Expressions involving operators and functions (e.g., to control the order of elements in a sequence).
- Conditional expressions that return boolean values from test conditions.
- Quantified expressions to test for the existence of some element that satisfies a condition.

The XML representation of an XQuery is defined in the W3C document called XQuery 1.0 and XPath 2.0 Functions and Operators. It was created by mapping the productions of the XQuery abstract syntax directly into XML productions. The result is not particularly convenient for humans to read and write, but it is easy for programs to parse, and because XQuery is represented in XML, standard XML tools can be used to create, interpret, or modify queries.

Note: For references to the documents listed in this and other sections of the specification see the Appendix.

2.2 Z39.50

There are two aspects to successfully using Z39.50:

- mastering the mechanics of the protocol (i.e., how are requests and responses created and deciphered) and understanding how to use the protocol (i.e., what search keys can be used at a particular site and what record formats do they support.)

Free and commercial Z39.50 client, server, and toolkit software is available for many programming environments. A large and incomplete list of such software is available at <http://lcweb.loc.gov/z3950/agency/resources/software.html>. Each implementation has its own API for creating Z39.50 requests and decoding Z39.50 responses. An alternate strategy for providing Z39.50 services over an existing database is to use a stand-alone Z39.50 server and export existing meta-data to this server.

In addition to the above website, there is a new initiative within the Z39.50 Implementers Group to create clients to a common object-oriented API. Information about the activities of this group can be found at <http://zoom.z3950.org/>.

Having established a Z39.50 session, the next step is to search the repository. The types of access points (e.g., author, title, etc.) are not mandated by the Z39.50 standard nor are the mappings from access points to fields in records (i.e., what fields in the record were indexed to provide the author access points). Such regulation is covered by application profiles. The Bath Profile (<http://www.nlc-bnc.ca/bath/bath-e.htm>) provides a description of how to do basic bibliographic searching and specifies mandatory record formats. It will be supported by most of the academic sites that

will serve as Z39.50 repositories to the IMS community. If the Bath Profile is deemed inadequate, then an IMS-specific profile could be created. Section 3.5 of the Best Practice Guide defines an experimental bib-1 Attribute set Use Attributes extension for IMS meta-data.

2.3 SOAP

SOAP is the Simple Object Access Protocol. It is currently a W3C Technical Report and work is being done to turn it into a W3C Recommendation. Information about SOAP can be found at <http://www.w3.org/TR/SOAP/>. The following paragraph is from the introduction provided on that page:

SOAP provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using XML. SOAP does not itself define any application semantics such as a programming model or implementation specific semantics; rather it defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems ranging from messaging systems to RPC.

SOAP, when combined with WSDL (Web Services Description Language <http://www.w3.org/TR/wsdl>), allows Web service providers to specify what services they can provide, what the inputs and outputs to those services are, and how to encode/decode the requests and responses exchanged between the clients and servers. SOAP and WSDL are programming language neutral and free toolkits for both clients and servers are available in a number of languages.

2.4 OAI Model Requirements

For the OAI model operations to work, it is necessary to fill in the date field in the associated meta-data. The <lom>.<metametadata>.<contribute>.<date><datetime> field should be complete as per the ISO 8601 recommendation. The date/time profile to be used for ISO 8601 is YYYY-MM-DDTHH:MM:SS. If any part of the time field is not relevant then it must set to '00'. Hence, a date only entry would be: YYYY-MM-DDT00:00:00. This date field must be updated C that the meta-data is changed.

3. Examples

The following examples of repository functions have been identified by the DRI Project Group and are currently being drafted to be included in the final specification prior to public release.

3.1 SOAP Messaging for XQuery Search Using IMS Meta-Data

The general message structure described in the DRI Binding document stipulates that the general SOAP message looks as follows:

```
Message
  header
  body
```

As discussed in that document, the header element is currently reserved and is an empty element. The body element is more fully described below:

```
message
  ...
  Message Body
    Message Type
    Payload Document
    Audit Elements
```

For an XQuery Search, the resultant SOAP binding of this model looks like the following:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope">
  <env:Header>
    <imsmsg:header xmlns:imsmsg= />
  </env:Header>
  <env:Body>
    <imsmsg:msgType>Search</imsmsg:msgType>
    <imsmsg:XQuery> </imsmsg:XQuery>
    <imsmsg:AuditElement> </imsmsg:AuditElement>

  </env:Body>
</env:Envelope>
```

For the XQuery results, the SOAP binding is similar and looks like the following:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope">
  <env:Header>
    <imsmsg:header xmlns:imsmsg= />
  </env:Header>
  <env:Body>
    <imsmsg:msgType>Search Results</imsmsg:msgType>
    <imsmsg:XQuery> </imsmsg:XQuery>
    <imsmsg:AuditElement> </imsmsg:AuditElement>

  </env:Body>
</env:Envelope>
```

The relevant schema for the messaging model is currently under development.

The XQuery syntax itself is that specified by the W3C. A schema is not currently available for the full XQuery syntax, In the context of the messaging model, it is therefore treated as plain XML. An example from the W3C specifications follows:

```
<bib>
{
  for $b in document("http://www.bn.com")/bib/book
  where $b/publisher = "Addison-Wesley" and $b/@year > 1991
  return
    <book year={ $b/@year }>
      { $b/title }
    </book>
}
</bib>
```

The results of an XQuery are themselves pure XML. For the above query, one possible response is:

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
  </book>
  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
  </book>
</bib>
```

3.2 Submit/Store Using FTP with IMS Content Packaging

The following are best practice recommendations for using FTP for the Request/Deliver and Submit/Store functionalities:

- 1) Only have read/write access for Submit/store function.
- 2) Only have read access for Request/Deliver function.
- 3) Binary transfer mode should always be used.
- 4) Control channel port: 21.
- 5) Recommended number of automatic connection retries: 12.
- 6) Recommended network timeout: 60 seconds.
- 7) If secure communications is required then various methods such as TLS/SSL may be used.

3.3 Z39.50 bib-1 Attribute Set Use Attributes Extension for IMS Meta-Data

Z39.50 allows private extensions to the Bib-1 Use Attribute set using values between 5,000 and 10,000. The following private extension to the Bib-1 Use Attribute set specifies an experimental attribute set defined for precise retrieval from IMS/LOM meta-data. Although the attribute set is defined for completeness, each implementation should determine the appropriate index for the meta-data profile selected.

Bib-1 Attribute Value	Attribute / Indexes LOM/IMS	Notes: Also indexes
	General	
5100	General- identifier -catalogue	General-catalogentry-catalogue (IMS)
5101	General-identifier -entry	General-catalogentry-entry (IMS) ALSO General identifier (IMS)
5102	General - title	
5103	General - language	
5104	General-description	
5105	General-keyword	
5106	General - coverage	
5107	General – structure - source	
5108	General – structure - value	
5109	General – aggregation level - source	
5110	General – aggregation level – source - value	
	Lifecycle	
5121	Lifecycle – version	
5122	Lifecycle – status - source	
5123	Lifecycle – status - value	
5124	lifecycle-contribute-role - source	
5125	lifecycle-contribute-role - value	
5126	lifecycle-contribute-entity	
5127	lifecycle – contribute - date	
	Metametadata	
5131	metametadata – identifier - catalogue	metametadata – catalogue entry – catalogue (IMS)
5132	Metametadata – identifier entry	metametadata-catalogue entry – entry AND metametadata-identifier (IMS)
5133	metametadata-contribute-role - source	
5134	metametadata-contribute-role - value	
5134	metametadata-contribute-entity	
5135	metametadata-contribute-date	
5136	metametadata-metadatascheme	
5137	metametadata-language	
	Technical	
5141	Technical - format	
5142	Technical – size	
5143	Technical – location	
5144	Technical – requirement	

Bib-1 Attribute Value	Attribute / Indexes LOM/IMS	Notes: Also indexes
5145	Technical – requirement– type - source	Technical – requirement – ORComposite – type – source (LOM)
5146	Technical – requirement – type - value	Technical – requirement – ORComposite – type – value (LOM)
5147	Technical – requirement – name - source	Technical – requirement – ORComposite – name – source (LOM)
5148	Technical – requirement - name - value	Technical – requirement – ORComposite – name – value (LOM)
5149	Technical – requirements – minimum version	Technical – requirement – ORComposite - minimumversion (LOM)
5150	Technical – requirement - maximum version	Technical – requirement - ORComposite maximumversion (LOM)
5151	Technical – installation remarks	
5152	Technical – other platform requirements	
5153	Technical – duration – date time	
5154	Technical – duration – description (IMS)	
	Educational	
5161	Educational – interactivity type - source	
5162	Educational – interactivity type - value	
5163	Educational – learning resource type - source	
5164	Educational – learning resource type - value	
5165	Educational – interactivity level – source	
5166	Educational – interactivity level - value	
5167	Educational – semantic Density - source	
5168	Educational – semantic Density - value	
5169	Educational – intended end user role – source	
5170	Educational – intended end user role - value	
5171	Educational – learning context - source	
5172	Educational – learning context - value	
5173	Educational – typical age range	
5174	Educational – difficulty - source	
5175	Educational – difficulty - value	
5176	Educational – typical learning time	
5177	Educational - description	
5178	Educational – Language (LOM)	
	Rights	
5181	Rights – cost - source	
5182	Rights – cost - value	

Bib-1 Attribute Value	Attribute / Indexes LOM/IMS	Notes: Also indexes
5183	Rights – copyright and other restrictions - source	
5184	Rights – copyright and other restrictions - value	
5185	Rights - description	
	Relation	
5190	Relation – kind	
5191	Relation – resource – identifier- catalogue	
5192	Relation – resource – identifier- entry	Relation – resource – identifier
5193	Relation – resource - description	
	Annotation	
5201	Annotation – entity	Annotation – person (IMS)
5202	Annotation – date	
5203	Annotation – description	
	Classification	
5211	Classification – purpose - source	
5212	Classification – purpose - value	
5213	Classification – taxonpath - source	
5214	Classification – taxonpath – taxon - Id	
5215	Classification – taxonpath – taxon - Entry	Classification – taxonpath – taxon (IMS)
5216	Classification – description	

The following Bib-1 Attributes should also be supported:

1011	Date Time added to Database
1012	Date Time last modified
1016	Any

Definition of a base attribute set is only one step in development of interoperable learning object discovery. A fully developed Z39.50 profile could also include:

- indexes which cross IMS attributes (e.g., general identifier index).
- indexing to other appropriate Bib-1 Use Attributes (e.g., where general identifier – catalogue = ISSN, index general- identifier – entry to 8).
- appropriate additional attributes and combinations of attributes to be supported.
- minimum Use attributes to be supported.
- record syntax support.
- Z39.50 services to be supported.

3.4 Cross-Domain Search Using IMS Meta-Data Access Points

To provide Cross-Domain Searching, meta-data search services could provide indexes which allow access to search clients from other domains.

DC	Use	IMS	Use	Bibliographic	Use
Identifier		General Identifier.Entry	5101 (also consider where catalog has specific values e.g., ISSN)	ISBN ISSN Identifier – Standard Identifier – document	7 8 1007 1032
Title	1097	General - Title	5102	Title	4
Language	1105	General – Language	5103	Code – Language	54
Description	1100	General - Description	5104	Abstract	52
Subject	1099	General Keyword, Classification with Classification Purpose equals “discipline” or “idea”	5105, 5215, 5217,	Subject Heading	21
Coverage	1110	General - Coverage	5106	Name Geographic	58
Type	1103	Educational – learning Resource Type	5164	Material Type	1031
Date	1102	Lifecycle – Contribute – Date where Contribute Role has value of Publisher.	5127	Date of Publication	31
Creator	1098	Lifecycle – Contribute – Entity where Contribute Role has value of Author.	5126 5125=Author	Author	1003
Other Contributor	1106	Lifecycle – Contribute – Entity with the type of contribution specified in Contribute Role.	5126 5126=role	Author	1003
Publisher	1101	Lifecycle – Contribute – Entity where Contribute Role has value of Publisher	5126 5126=Publisher	Publisher	1018
Format	1107	Technical - Format	5141	Material Type	1031
Rights	1111	Rights – Description	5185	Note	63
Relation	1109	Relation – Resource Description	5193	Note	63
Source	1108	Relation – Resource where the value of Relation Kind is “based on”	5191, 5192	Note	63
Audience		Educational – intended end user role, Educational – learning context – value, Educational – typical age range	5170, 5172, 5173	Note	63

3.5 Learning Objects Network Example

The following is a description of a working testbed for a network of distributed repositories using SOAP-based messaging from Learning Objects Network, Inc. (<http://www.learningobjectsnetwork.com>).

Learning Objects Network, Inc. (LON) is developing a testbed implementation of a network of distributed repositories and at the January 2002 IMS meetings in Cambridge, MA, demonstrated a working model using messaging and meta-data search capabilities like that recommended in the IMS Digital Repository Base documents.

The demonstration included 2 sample repositories containing SCORM 1.2 conformant learning objects, an XML repository containing the IMS/SCORM 1.2 meta-data for each of the learning objects, an XQuery-based search tool, a Digital Object Identifier Registry Server, a commercial LMS, and SOAP-based messaging to permit search and retrieval of learning objects across the network.

The demonstration used the Learnframe LMS (a pre-release version which had implemented the messaging) running on a notebook computer. A user inside Learnframe could construct a simple query which was passed by the messaging to the XQuery engine running on a computer with the XML meta-data repository. All matching meta-data were returned to Learnframe where the user could examine the meta-data and choose objects of interest. Once an object was chosen, the application took the unique ID for the Object (a DOI) and performed a lookup at the LON DOI Registry Server which returned information on which repository contained that particular object. Learnframe then sent a request to the repository to deliver the object. The learning object (or SCO) was delivered to Learnframe and run inside the LMS.

This example serves as an early proof-of-concept for use of XQuery as a search tool against the full IMS meta-data, for the use of an XML repository to aggregate meta-data from multiple repositories, for the SOAP-based messaging to search and retrieve standard learning objects as implemented by multiple vendors' products, and for the use of the DOI as a globally unique identifier for resolving the location of the learning objects in a network of distributed repositories.

Since this demonstration, the prototype has been extended with the IDL iDesigner authoring tool which implemented the messaging model. This allowed the course author to select and retrieve multiple learning objects and assemble them into multiple small courses, as well as demonstrate dynamically adaptive learning that would alter the path through the SCOs based on the user's learning styles.

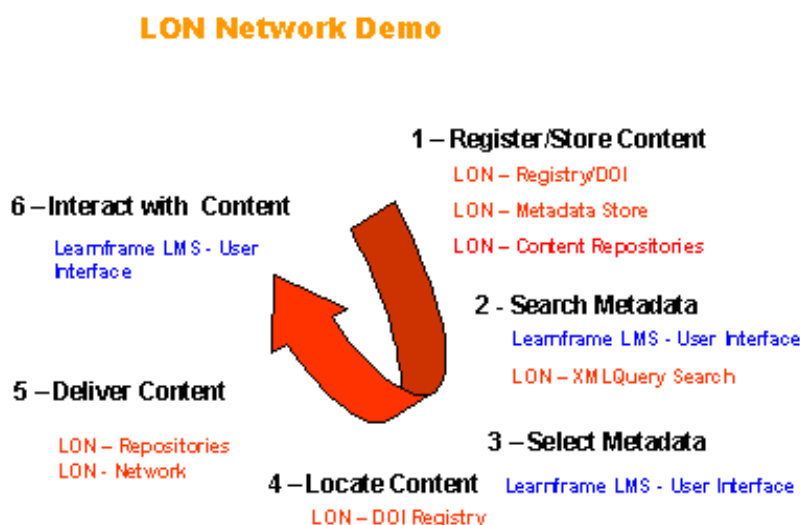


Figure 3.1 Diagram of Learning Objects Network Working Prototype for search and retrieval of learning objects on a network of distributed repositories.

4. Future Consideration

This section has been included in order that technologies and issues, which are beyond the scope of the current specification, can nevertheless be raised and some indication given of the group's thinking on these topics. These considerations are as much an invitation for the reader to contemplate these issues and provide feedback to the group, as they are an indication of what is under consideration for the future.

4.1 Provisions and Disclaimers

It should be noted that inclusion of a topic in this section is in no way a commitment that it will subsequently be included in a future version of a DRI or more general IMS specification. These are indicative only, and may be subject to change due to external events as well as by decisions by the DRI Project Group. Matters to be included in the scope of future specifications will be determined by priorities and the stability of technologies under consideration at that time.

4.2 Registries and Directories

The requirements for directory and registry services are not identified in detail at this time. Therefore, no specific technology recommendations are made in this section.

The notions of directories and registries are merging into a continuum. A classic definition of a directory or directory service as found in the ISO X.500 recommendation is "the Directory is a collection of open systems which cooperate to hold a logical database of information about a set of objects in the real world." This notion was extended in a subsequent Microsoft definition to be "[...] repositories for information about network-based entities such as applications, files, printers and people." A contemporary description of UDDI is wholly compatible with the above definitions: "A common analogy used for UDDI is a phone book for Web services. It has business names, business mailing addresses, contact names, contact phone numbers, Web services offered by businesses, addresses of Web services, meta-data describing the 'interfaces' of Web services, etc." For our purposes, we might amalgamate these definitions into the following:

"A directory or registry is a repository for information about network-accessible entities, such as services, repositories, specifications, content objects, people, and so on."

In the context of this document, we make no firm distinctions between these two terms, *directory* and *registry*, and choose rather to characterize the classes of directories and registries based on their required behaviors.

4.2.1 Identify and Locate Services

The emerging view of network-centric application models is of a set of network-accessible services that publish their descriptions in a consistent fashion to well-known network locations. This allows other applications or services to programmatically discover, locate, and interoperate with these services.

A number of specifications for this model or implementations of this model exist, including:

UDDI (<http://www.uddi.org/>)

"Universal Description, Discovery and Integration (UDDI) is a specification for distributed Web-based information registries of Web services. UDDI is also a publicly accessible set of implementations of the specification that allow businesses to register information about the Web services they offer so that other businesses can find them."

JXTA (<http://www.jxta.org/>)

"JXTA technology is a set of open, protocols that allow any connected device on the network ranging from cell phones and wireless PDAs to PCs and servers to communicate and collaborate in a P2P manner." Of particular interest to this community is the notion that JXTA allows us to "find peers and resources on the network."

4.2.2 Identify and Locate Objects

Services operate on objects. Therefore, for a distributed network-based application model, it is just as important to be able to discover, obtain rights to, and use objects, as it is to do discover the services that operate on them. A large number of repositories for such objects exists in various communities of interest today. Of particular interest in the context of this document are the methods used to identify and locate objects within these repositories.

DOI (<http://www.doi.org/>)

The DOI is a GUID that allows resolution of multiple instances of object location and is a permanent identifier. Resolution is used to mean the act of submitting an identifier to a network service and receiving in return one or more pieces of current information related to the identifier (including the location of the object).

OpenURL (<http://www.sfxit.com/openurl/openurl.html>)

The OpenURL is a method of reference linking which is gaining widespread acceptance in the publishing and library communities. Rather than seeking to be independent of physical location, the advantage of OpenURL resolution is in finding the appropriate copy or copies of an item which are stored in multiple locations.

PURL (<http://www.purl.org/>)

The Persistent Uniform Resource Locator (PURL), provides a method of resolving URLs through the use of an intermediate resolution service rather than pointing directly to a Web address or Internet resource.

4.2.3 Identify and Locate People

Many of the services in the e-learning context affect or interact with people. It is therefore important to be able to identify and locate information that relates to the individuals in question. A specific example of this is the need to access and track learning profiles for individuals. There are many registries that enable the location of people today. These are mostly contained within the boundaries of specific organizations, may have restricted access, and implement one or both of the dominant access models: X.500 and LDAP.

In the narrower context of e-learning, information about people is identified and can be located, but is usually maintained within the confines of a single e-learning solution (e.g., an LMS). No generalized standards or implementations of public registries for such profile information are currently identified.

X.500 (<http://www.iso.ch/>)

X.500 can be used to build a standards-based directory, than can access information in a uniform manner, no matter where the user is based.

LDAP (<http://www.ietf.org/>)

Lightweight Directory Access Protocol (LDAP) is a technology for accessing common directory information. LDAP provides an extendable architecture for centralized storage and management of information that needs to be available for today's distributed systems and services.

4.2.4 Identify and Locate Resources

Many different types of more generalized resources may be needed to allow a distributed network model to interoperate successfully. One common example of such resources includes published and publicly accessible XML schemas, such as those issued by the IMS itself, the XML.org specifications registry, and so on.

IMS (<http://www.imsglobal.org/>)

IMS Global Learning Consortium, Inc. (IMS) is developing and promoting open specifications for facilitating online distributed learning activities such as locating and using educational content, tracking learner progress, reporting learner performance, and exchanging student records between administrative systems.

XML.org (<http://www.xml.org/xml/registry.jsp>)

The XML.org Registry is a community resource for accessing XML specifications being developed for vertical industries and horizontal applications. The XML.org Registry is a central clearinghouse for developers and standards bodies to publicly submit, publish, and exchange XML schemas, vocabularies, and related documents.

4.3 Digital Rights Management

Digital Rights Management (DRM) refers to the problem of managing the intellectual property rights over assets including identifying rights holders, the applicable allowable permissions, and tracking usage. DRM has traditionally been focused on security and encryption as a means to solve these issues. That is, lock the content and limit its distribution to only those who pay. This is what is being referred to as “first-generation DRM” and represents a substantial narrowing of DRM’s real and broader capabilities. DRM is now being defined to cover the description, identification, trading, protection, monitoring, and tracking of all forms of rights permissions, constraints, and requirements over both tangible and intangible assets including management of rights holder’s relationships. This is the “second-generation DRM”.

Additionally, it is important to note that DRM is the “digital management of rights” and not the “management of digital rights”. That is, DRM manages all rights, not just the rights only applicable to permissions over digital content.

The overall DRM framework suited to support the philosophy in building digital rights-enabled systems can be modelled into three areas:

- **Create:** How to manage the creation of content so it can be easily traded. This includes asserting rights when content is first created (or reused and extended with appropriate rights to do so) by various content creators/providers.
- **Trade:** How to manage and enable the trade of content. This includes accepting content from creators into an asset management system. The trading systems need to manage the descriptive meta-data and rights meta-data (e.g., parties, usages, payments, etc.).
- **Use:** How to manage the usage of content once traded. This includes supporting constraints over traded content in specific desktop systems/software.

4.4 Location and Resolution Services

The Request/Deliver component of the DRI model is based on a pointer within a meta-data record or alert message that references the location of the digital object. The IMS Meta-Data Specification makes provision for this pointer to take the form of a method which resolves to a location.

The three main methods of resolution that have potential application here include DOI, OpenURL, and PURL.

4.4.1 DOI

The DOI syntax is described in the ANSI/NISO Standard Z39.84-2000 and conforms to the IETF RFC 1737 Functional Requirements for Uniform Resource Names (<http://www.ietf.org/rfc/rfc1737.txt>). With the use of a proxy server, the DOI may also be resolved using the common URL structure. For example, “doi:10.1000/123” may be resolved using “<http://dx.doi.org/10.1000/123>”.

The DOI was developed by the publishing industry and is now maintained by the International DOI Foundation, a non-profit organization. It has wide support in the publishing sector and is already in commercial use for Journal references (Cross-Ref) and eBooks. The DOI is a GUID that allows resolution of multiple instances of object location and is a permanent identifier. Resolution is used to mean the act of submitting an identifier to a network service and receiving in return one or more pieces of current information related to the identifier. In the case of the Domain Name System (DNS), as an example, the resolution is from domain name (e.g., www.doi.org), to a single IP address (e.g., 132.151.1.146), which is then used to communicate with that Internet host. In the case of the DOI System, the resolution is from a DOI (e.g., 10.1000/140), to one or more pieces of typed data (e.g., three URLs representing three copies of the object). With the DOI in the IMS meta-data, it is possible to locate the object now and in the future, despite changes in ownership, etc.

Further information about DOI is available at <http://www.doi.org>, and about the ANSI/NISO standard at http://www.niso.org/standards/standard_detail.cfm?std_id=480.

The DOI is a fee-based service.

4.4.2 OpenURL

The OpenURL is a framework for open and context sensitive method of reference linking that is gaining widespread acceptance in the publishing and library communities. Rather than seeking to be independent of physical location, the advantage of OpenURL resolution is finding the appropriate copy or copies of an item that are stored in multiple locations. See <http://www.sfxit.com/openurl/openurl.html> for more information. Although OpenURL has been developed in the context of Scholarly literature, a framework for generalizing the model to other domains has been put forward (the 'Bison-Futé' model - see <http://www.dlib.org/dlib/july01/vandesompe/07vandesompe.html>). This generalized model could be used as the basis for adoption within the IMS DRI community.

OpenURL has been fast-tracked by NISO for adoption as a NISO standard. Committee AX is taking this forward (see <http://www.niso.org/commitax.html> for information on progress).

An OpenURL is basically an HTTP request with embedded meta-data that conforms to a prescribed syntax. The syntax of this 'hook' is basically a series of concatenated name=value pairs with the <searchpart> of a URL expressed in a standard `http://<host>:<port>/<path>?<searchpart>` format.

The resolution service takes the <searchpart> and resolves it to point to third-party services that hold the referenced item and are appropriate to the context of the user.

The 'Bison-Futé' model allows for pointers to be set up in such a way that the need for such intelligence in the resolution service is eliminated. In this model, the referenced item (e.g. the Learning Object) is called a *referent*, the standardized reference is known as a *descriptor*, and the hook which cites the reference is called a *ContextObject* as opposed to OpenURL.

Within the IMS DRI context it is useful to have meta-data that separates out the generic information about an object from the service component that points to where appropriate, additional, or alternative instantiations of the object may be located for a particular user. Any resolver service would therefore need to interface closely with the digital rights management component as well as the request-deliver line in the framework document.

Recent work has attempted to bring the OpenURL and DOI initiatives together. See <http://www.dlib.org/dlib/september01/caplan/09caplan.html> for details of a prototype system. This initiative is worth tracking as it may form the best basis for adoption in future as a standard for the IMS community.

4.4.3 PURL

PURL provides a method of resolving URLs through the use of an intermediate resolution service rather than pointing directly to a Web address or Internet resource. The PURL resolution service associates the PURL with the actual URL and returns that URL to the client. The client can then complete the URL transaction in the normal fashion, similar to the HTTP redirect function. While the PURL remains forever the same, the URL associated with it can be changed and updated as needed. See <http://www.purl.org/> for more information.

PURL does not restrict the use of this service and anyone can host a PURL resolver.

4.5 Request/Deliver Services

The specification for the Request/Deliver interface included in version 1.0 of the DRI Information Model includes provision only for requesting online learning objects from a learning object repository where authorization has already taken place and verification of delivery is not required.

In a hybrid environment where search agents are accessing a range of repositories, it would be useful to be able to use a common system for requesting items that:

- Are not available online.
- Are not available in digital forms.
- Require off-line authorization (e.g., approval by a faculty member or librarian) before submission.
- Require more complex messaging between requester and supplier (e.g., confirmation of delivery, cost negotiation).

A useful starting point for integrating these services with the Request/Deliver service specified in the DRI Information Model is ISO 10161 for inter-library loans (see <http://www.nlc-bnc.ca/iso/ill/stan1611.htm>). The state model for ISO 10161 is very comprehensive and the IMS community may only require a subset of the messages and message transitions defined in the protocol.

The ISO 10161 Protocol Implementer's group (IPIG) has also defined an XML schema for a 'patron-request' message. This could be used to cover the scenario listed above where a student's request for an item needs to go to a third-party system for mediation prior to submission. The schema can be viewed at http://www.nlc-bnc.ca/iso/ill/document/ipigwp/ipigrequestsubmission_xsd.html.

4.6 Web Services

Web Services are currently deployed in a variety of proprietary and non-proprietary ways. During 2002, the W3C established a number of working groups to develop abstract models and formal definitions of Web Services that are independent of underlying transport protocols, operating systems, and programming languages. The Web Services Description Language (WSDL) has been one of the ongoing outputs of this work and is a means of exposing services and middleware applications and data in a standard XML description.

WSDL is thus positioned to support the interactions of distributed programming technologies such as CORBA and Enterprise Java Beans and B2B Document Flows (e.g., RosettaNet and ebXML).

The concept behind WSDL--being able to invoke remote functionality--is not new in the computer industry. CORBA, DCOM, and RMI currently are successful in doing this in tightly managed environments. The key difference with the WSDL approach is that it is intended to work over the Internet and provide support for a multitude of underlying service protocols and interface options.

The following is a description taken from the current W3C WSDL Note available at <http://www.w3.org/TR/wsdl>:

"WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME."

However, in the context of ongoing IMS specification work relevant to Digital Repositories Interoperability, WSDL is still considered to be under development despite its growing popularity. The Web Services Description Working Group (<http://www.w3.org/2002/ws/desc/>) has been chartered to progress the W3C WSDL Note into a formal W3C Recommendation. This WG is scheduled to complete this task by May 2003.

4.7 Recommendations Regarding GUID Allocation

There are numerous IMS specifications making reference to unique identifiers for resources that can be stored in digital repositories. These specifications include the meta-data reference to identifier elements, Content Packaging, the Learner Information Profile Specification and others. Recognizing the question of Globally Unique Identifiers (GUIDs) to be a significant one, the IMS has produced an Implementation Handbook on the subject, which provides a strawman base proposal for the use of GUIDs in implementations of IMS specifications (IMS, 2001). Because this strawman base proposal reflects current thinking in IMS across a number of working groups regarding GUIDs, the Digital Repository Interoperability Specification refers its readers to this document, and will summarize its recommendations very briefly.

The Implementation Handbook proposes the IETF Uniform Resource Name (URN) as a candidate for a future IMS UID scheme, and it describes the URN as follows:

“The URN syntax is: URN:[<nid>]:[<nss>] where [<nid>] is the Namespace Identifier and [<nss>] is the Namespace Specific String. The namespace is a collection of uniquely-assigned identifiers. A URN NID is used to ensure global uniqueness of URNs and possibly provide a mechanism to decode the structure of the identifier. An organization may apply for a (formal or informal) NID from the IANA. The namespace string is the unique, persistent object ID within a namespace.”

From the IMS Persistent, Location-Independent, Resource Identifier Implementation Handbook (http://www.imsglobal.org/implementationhandbook/imsrid_handv1p0.html).

5. Support for Roles

The DRI specification outlines use cases that illustrate the set of roles adopted most commonly by the users (including both people and software) of the e-learning application, digital repositories, and information services. The Role undertaken at any time is dependent on the context. The users may adopt more than one role over the life of an event including interactions with repositories.

5.1 Learner

A Learner is defined as a person following a learning path and/or enrolled in a course or training program. The Learner can be actively engaged in a learning activity delivered as part of an e-learning application. From within the application, a Learner may need to discover resources that are required to complete an assignment, or may facilitate the completion of a learning task—a discussion, quiz, group project, or chat are examples of learning tasks that happen within an e-learning environment. The Learner may be affiliated with either a learning application or a learning information service, acting on his/her own behalf in the pursuit of knowledge. Once the Learner leaves the e-learning application, his/her primary role may change from a Learner to an Infoseeker.

5.2 Creator

A Creator is defined as a person responsible for the creation of learning objects or other resources, or the construction of learning paths, courses, or learning programs. A Creator may be an independent content developer, content publisher, instructional designer, manager, instructor, or tutor that may be collecting, constructing, and sequencing content for delivery to a Learner. A Creator is someone within an e-learning environment or information services.

5.3 Infoseeker

An Infoseeker is defined as a person seeking to obtain information through the discovery of resources. A Learner may become an Infoseeker in order to complete tasks during a course or training program. A Creator may also become an Infoseeker, discovering and accessing resources of all kinds, including learning objects. An Infoseeker does not have to be inside an e-learning system, and may be accessing resources from a library or other information services located within the enterprise.

5.4 Agent

An Agent is defined as an intelligent software application that carries out tasks directly on behalf of a Learner, Creator, or Infoseeker. The Agent could be engaged from inside the e-learning application, digital repository, or information services. The results of the Agent may be processed programmatically or with intervention from the Learner, Creator, or Infoseeker.

6. Conformance Issues

6.1 No Conformance Statement

It is inappropriate to define a 'Conformance Statement' for the DRI specification. The DRI describes a number of alternative ways in which digital repositories can exchange information i.e., there is no single preferred approach. Each solution can have almost any number of subtle differences and so there are, in effect, an infinite number of possible solutions that would comply with the DRI recommendations. It is for this reason that it is inappropriate, if not impossible, to make a single conformance statement.

Instead, we will look at the details that need to be defined in order to create a system description based upon the DRI recommendations. Once these details have been defined for a particular architecture, it should then be possible for the corresponding conformance statement to be written for that architecture.

6.2 Application Profiling

The DRI specification references the following specifications and standards:

- Digital Object Identifier (DOI)
- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP) and Secure HTTP (HTTPS)
- IEEE Learning Object Metadata (LOM)
- IETF URN
- IMS Content Packaging
- IMS Meta-Data
- IMS Persistent Location-independent Resource Identifier
- Lightweight Directory Access Protocol (LDAP)
- Open Archive Initiative Protocol for Metadata Handling (OAI-PMH)
- OpenURL
- Simple Mail Transfer Protocol (SMTP)
- SOAP and SOAP with attachments V1.1
- Universal Description, Discovery and Integration (UDDI)
- X500 Directory Service
- XPath V2.0
- XQuery V1.0
- Z39.50

The combination of the above specifications and standards depends on the actual DRI implementation. The DRI provides a number of recommendations to promote interoperability however the realization of interoperability requires further system definition. IMS terms this system definition the Application Profile (AP). The creation of any DRI system using an AP must define the combination of specifications and/or standards that are to be adopted. Once these have been identified the following profiling of each specification and standard is required:

- Identification of the optional characteristics within a specification/standard that are mandatory in the AP;
- Identification of any ordering requirements in the information files - many specifications support unordered data sets however many implementations prefer a particular order;
- Selection of the appropriate profiles when a variety are supplied e.g. the definition of one format from the many support for date/time in ISO 8601;

- The identification of the vocabularies that are to be used and the definition of which entries in a vocabulary are to be used in each circumstance;
- The definition of the extension functions that are to be used to overcome any functional deficiencies in the specification/standard;
- The imposition of data typing when possible and where appropriate e.g., many data fields are given as strings when in fact they can be typed e.g., a URL;
- The definition of the min/max constraints to ensure that systems can exchange/store a minimum amount of data;
- The definition of the appropriate naming conventions and unique identifier conventions.

6.3 Implementation Issues

The detailed implementation guidelines that need to be considered when defining the AP include:

- SOAP messaging - most SOAP implementations are 95% + interoperable. However, there are incompatibilities e.g., handling of the 'must-understand' option;
- The model for digital repositories - it is common practice in digital repositories to separate the storage aspects from the search capabilities. Therefore, a repository must respond to requests to store, retrieve and delete objects using an agreed-on unique identifier for the object;
- The search model - the DRI Search model will handle any type of IMS Meta-Data i.e., differences created due to the use of extensions. Therefore, any DRI Search model needs to address extensibility (reasonably) well. Though a formalism exists to map any hierarchical structure into a relational model, and even in principle further to a 'flat' model, this is not very workable in practice. This means that DRI search must handle metadata documents natively. Since XML is the chosen initial expression for these documents, it follows that DRI search must at least handle XML natively. This does not exclude support for other search models - i.e., Z39.50 - it merely states a necessary condition that IMS DR Search must satisfy;
- A single XML document may be searched using XPath to query its tree representation. Useful search must address searches over collections of documents. The current XML Query 1.0 and XPath 2.0 Draft [XQUERY] appears to address this need. Any query solution should also support relational data, which the use cases for this specification do. However, [XQUERY] only defines semantics and a data model, specifically not syntax (there is a working draft for an XML Syntax for XQuery (XQueryX) that does). It is a good idea to define Search at least at the XQueryX syntax model level;
- Security - there needs to be some minimal level of both user- and server-side authentication for any 'real' implementation.

Appendix A – Glossary

Access	An action (such as a query or direct hyperlink) by either a human or machine enabling the retrieval of data.
Agent	Software or a system component designed for a specific, automated task such as the discovery of <i>meta-data</i> or to <i>request assets</i> .
Alert	A message instigated from a <i>repository</i> and received by a <i>resource utilizer</i> often to inform a <i>resource utilizer</i> of new or updated <i>meta-data</i> and/or <i>resources</i> .
Asset	Any digital <i>resource</i> that can be submitted to or requested from a <i>digital repository</i> . In an IMS context, such a resource is typically used in instruction or learning, and is either purposed as a learning object, a component of a learning object, or a ‘raw’ information resource. While <i>meta-data</i> can also be utilized as an asset, the information model distinguishes between assets and their associated meta-data.
Creator	An entity primarily responsible for making the content of a <i>resource</i> .
Deliver	To transport an <i>asset</i> from a <i>repository</i> to a <i>resource utilizer</i> in response to a <i>request</i> .
Digital Repository	A collection of digital <i>assets</i> and/or <i>meta-data</i> accessible by <i>resource utilizers</i> via a network without prior knowledge of the digital repository’s structure.
Directory	Typically a <i>digital repository</i> of information about network-accessible entities such as services, other repositories, specifications, content objects, people, and organizations that is managed, searchable, and organized according to a specific logic (such as alphabetical order).
Discover	An action resulting from a <i>query</i> that involves presentation of results to the <i>user</i> . Querying typically depends on <i>meta-data</i> being exposed for effective discovery. Querying, browsing, ‘following a path’ are all aspects of the discover function, commonly called discovery.
Expose	To make <i>meta-data</i> available to <i>resource utilizers</i> for purposes of <i>searching</i> , <i>gathering</i> , and <i>alerting</i> .
Gateway	Typically, a functional interface to a <i>digital repository</i> that provides pathways to either single or multiple search requests and results. Translations of syntax may be performed by a gateway.
Gather	To identify, transport, and process <i>meta-data</i> that is <i>exposed</i> by a <i>digital repository</i> for the purposes of resource utilization and/or re-purposing. The <i>meta-data</i> is commonly gathered as a bulk collection.
Gather Engine	A software application commonly known as a Harvester.
Infoseeker	Any <i>user</i> or <i>agent</i> attempting to <i>discover</i> , <i>access</i> , <i>search</i> , or <i>gather</i> assets.
LCMS	A Learning Content Management System (LCMS) is a multi-user environment where learning developers can create, store, reuse, manage, and deliver digital learning content from a central object repository. LCMS products allow users to create and reuse small units of digital learning content. An LCMS manages the process of creating and delivering learning content, just as the words indicate.
Learner	Any user who uses <i>resources</i> to gain knowledge, understanding, or skill.
LMS	A Learning Management System (LMS) is a computer application that enables learning management.
Manage	Functions related to the internal maintenance and operation of a <i>digital repository</i> .
Meta-data	Descriptive information about data that typically supports operations on <i>assets</i> . It is structured data used to describe the characteristics of a learning resource, a data object, or a component of a learning technology system.

Publish	To transport an <i>asset</i> from a <i>creator</i> to a <i>digital repository</i> for storage.
Query	Operation performed typically by a <i>user</i> or <i>agent</i> wishing to <i>discover</i> and have information delivered from a system.
Registry	A system that typically holds meta-data schemas, configuration data, application profiles, or other lookup data that is both human- and machine-accessible. A registry can also be understood as an instance of a <i>digital repository</i> .
Request	To ask a <i>digital repository</i> for the <i>delivery</i> of an <i>asset</i> .
Resolver	A service that enables access to a specific instance of a <i>resource</i> .
Resource	Any digital <i>asset</i> that can be accessed to support learning. A resource can be a meta-data record, a learning object, a component of a learning object, or any digital asset.
Resource Utilizer	Any functions that make use of <i>assets</i> or <i>meta-data</i> .
Search	Operation performed by a <i>user</i> or <i>agent</i> using <i>meta-data</i> and the <i>expose</i> function typically prior to retrieving desired <i>assets</i> .
Search Model	A defined method for searching the <i>meta-data</i> .
Spider	A software application used by search engines to index website content. Also commonly referred to as a 'crawler'.
Store	To receive and maintain an <i>asset</i> in a persistent form upon its publication.
Submit	To transport an asset from a <i>creator</i> to a <i>digital repository</i> for storage.
User	In a broad sense, the roles of <i>learner</i> , <i>creator</i> , or <i>infoseeker</i> .

Appendix B – Possible Future Directions

This appendix describes the current thinking on the Alert/Expose function identified in the DRI Information Model. Alert services are in common use in repositories (e.g., library information services), but currently there is no common mechanism for propagating Alerts across service providers. Alert/Expose is a core function for a repository, but this area requires further work before an interoperable solution can be proposed.

The DRI Project Group regards the Alert function as a possible component of a digital repository or an intermediary aggregator service and envisions that email/SMTp (Simple Mail Transfer Protocol) could provide this functionality. However, the Alert function is regarded as out of scope for Phase 1 of the DRI specification.

The following is a detailed description of how the Alert process may work in a future version of the specification that includes this functionality.

The Alert reference model adds to and is dependent on the Search reference model. A diagram of this model is shown in Figure B.1 below.

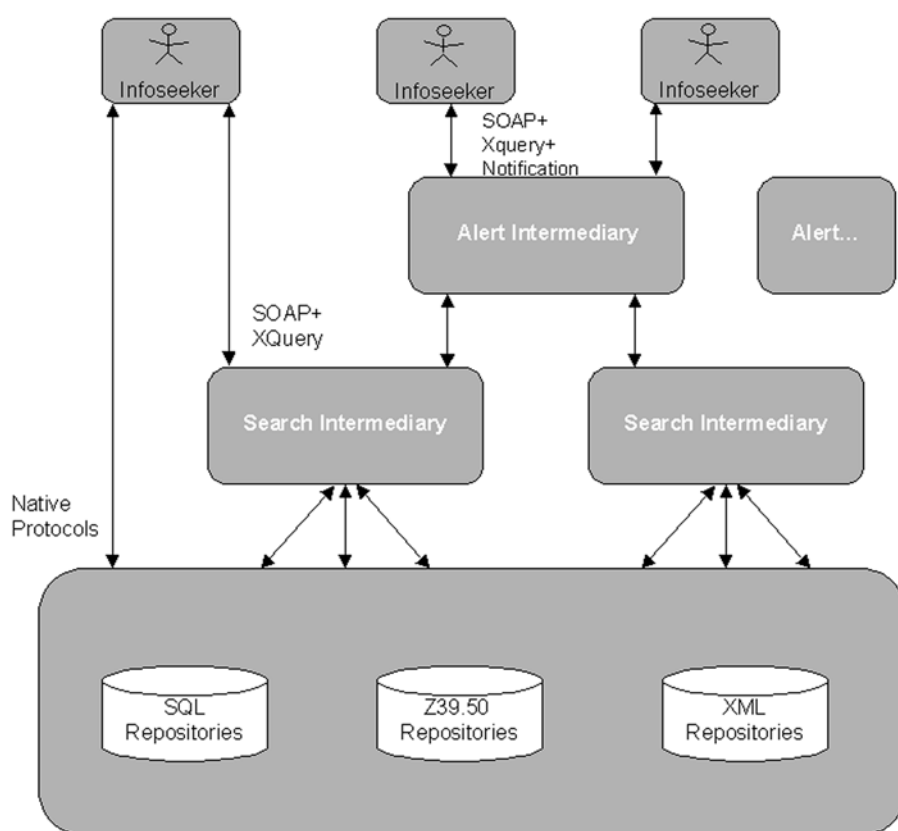


Figure B.1 Diagram of the Alert reference model.

The Alert model includes the Alert intermediary module(s) that act on behalf of the infoseeker and manage the Alert requests.

An Alert request is composed of the following:

- Defined event that triggers the alert
- Frequency to check the event
- Delivery option for the alert results

The defined events are specified with an XML query that indicates the query to be executed over the repository. The XML query also defines the elements in the query result, which is also the delivery result to be sent to the infoseeker.

The frequency of the event to be queried is expressed as a period of time (defined by ISO8061). The event query must be executed at the end of this period. The frequency may also optionally indicate the maximum number of times to query the event. If there are no results from the event query, then notification is not required. If there are any results from the event query, then an appropriate notification must be delivered. Additionally, the infoseeker may also request to be notified irrespective of the outcome of the event query results.

The delivery options indicate the method in which the infoseeker requests the notification to be delivered. Current delivery options include email using SMTP.

The remainder of the Alert model is the same as the Search model. The Search model will accept requests from the Alert intermediary (just as if it was a infoseeker) and process the query as per the functions of the Search intermediary module(s). The results are sent back to the Alert intermediary for it to process and notify/deliver to the infoseeker.

Alert Binding

The <alert> element must contain the following three children elements when submitting an alert request:

- <event>
- <frequency>
- <notification>

The <alert> element also contains an “id” attribute (a URI value) that contains a unique identifier for this alert request.

The <event> element must contain a valid W3C XQuery that is to be executed to meet the infoseeker’s requirements. The <event> element’s XML Schema is defined to contain any elements from the XQuery XML namespace. The elements must be validated to conform to the XQuery rules.

The <frequency> element contains two children elements:

- <period>
- <repeat>

The mandatory <period> element contains a valid period of time defined by [ISO8061]. The <period> element is defined with the “xsd:duration” datatype from XML Schema.

The optional <repeat> element contains a positive integer value indicating the maximum number of times the query should be executed. If there is no <repeat> element, then the query should be repeated indefinitely.

The <frequency> element also supports a boolean attribute called “all-notify”. If “true”, then the infoseeker will always be notified even if there are no positive results from the query. If “false”, the default value, then the infoseeker will only be notified when there are positive results from the query.

The <notification> element contains two children elements:

- <method>
- <method-data>

The <method> element indicates the system used to notify the infoseeker from the results of the Alert request. There is currently only one valid enumerated value for this element:

- email

The <method-data> element indicates additional information that is required for the system defined in the <method> element. For the case of “email”, a valid email address must be supplied in the <method-data> element.

The <alert> element must contain the following element when returning the results of an alert request:

- <results>

The <results> element contains the XML elements defined as the output results from the XML query. The <results> element also has an “id” attribute that links it to the original alert request.

All of the above <alert> information is transported as a SOAP payload.

Alert/Expose Example

The following shows an example Alert request. The query is executed every 14 days for a maximum of 50 times. The results are sent via email.

```
<dr:alert id="urn:example.com:alerts:3737372:ZZ22"
  xmlns:dr="http://imsproject.org/schema/dr/2001"
  xmlns:xq="http://www.w3.org/2001/06/xquery">
  <dr:event>
    <xq:query>
      <xq:flwr>
        ...
      </xq:flwr>
    </xq:query>
  </dr:event>

  <dr:frequency all-notify="false">
    <dr:period> P14D </dr:period>
    <dr:repeat> 50 </dr:repeat>
  </dr:frequency>

  <dr:notification>
    <dr:method> email </dr:method>
    <dr:method-data> me@example.com </dr:method-data>
  </dr:notification>
</dr:alert>
```

The following shows an example of the response from the above alert request:

```
<dr:alert xmlns:dr="http://imsproject.org/schema/dr/2001"
  xmlns:ims="http://imsproject.org/schema/md/2000">
  <dr:response id="urn:example.com:alerts:3737372:ZZ22">

    <ims:title> ... </ims:title>
    <ims:contribute> ... </ims:contribute>

  </dr:response>
</dr:alert>
```

Appendix C – Additional Resources

XQuery References

XML Query 1.0: An XML Query Language

World Wide Web Consortium. XML Query, Version 1.0. W3C Working Draft, 15 November 2002.

<http://www.w3.org/TR/xquery/>

XQuery 1.0 and XPath 2.0 Formal Semantics

World Wide Web Consortium. XQuery 1.0 and XPath 2.0 Formal Semantics. W3C Working Draft, 15 November 2002.

<http://www.w3.org/TR/query-semantics/>

XML Query 1.0 and XPath 2.0 Data Model

World Wide Web Consortium. XML Query 1.0 and XPath 2.0 Data Model W3C Working Draft, 15 November 2002.

<http://www.w3.org/TR/query-datamodel/>

XML Query 1.0 and XPath 2.0 Functions and Operators

World Wide Web Consortium. XML Query 1.0 and XPath 2.0 Functions and Operators W3C Working Draft, 15 November 2002.

<http://www.w3.org/TR/xquery-operators/>

About This Document

Title	IMS Digital Repositories Interoperability - Core Functions Best Practice Guide
Editors	Kevin Riley (IMS), Mark McKell (IMS)
Team Co-Lead	Jon Mason (IMS Australia - DEST)
Version	1.0
Version Date	13 January 2003
Status	Final Specification
Summary	This document describes the Best Practice for implementing the IMS Digital Repositories Interoperability Specification.
Revision Information	13 January 2003
Document Location	http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_bestv1p0.html

List of Contributors

The following individuals contributed to the development of this specification:

Name	Organization
Tom Barefoot	Learning Objects Network
Fred M. Beshears	UC Berkeley
Kerry Blinco	IMS Australia - DEST
Dipto Chakravarty	Artesia Technologies
Tracy Flynn	Learning Objects Network
Norm Friesen	Industry Canada
Niccolo' Giaccone	Giunti Labs
Renato Iannella	IMS Australia - DEST
Kirk Johnson	ADL
Paul Lefrere	Open University
Ralph LeVan	OCLC
Esko Liimatta	R5 Vision
Aki Salakka	R5 Vision
Rick Scurfield	Artesia Technologies
Neil Smith	FDI
Colin Smythe	IMS
Mats Svensson	Luvit
Kim Voltero	WebCT
Darrell Woelk	Docent

Revision History

Version No.	Release Date	Comments
Base 1.0	05 February 2002	The first formally released version of the full IMS Digital Repositories specification.
Public Draft 1.0	16 July 2002	Expanded several sections and edited for consistency.
Final 1.0	13 January 2003	Made minor edits to clarify meaning and provide better textual description in various sections. Corrections to minor typographic errors were also included. <ul style="list-style-type: none">- Added section 2.4 "OAI Model Requirements".- Added section 6 "Conformance Issues".

Index

A

agent 17, 20, 23, 24
 aggregator 25
 Application Profile 21, 22
 Architecture 3
 architecture 15
 Asset 23
 asset 16, 23, 24

C

conformance 21
 Content Packaging 3, 18
 core functions 3
 creator 12, 16, 20, 23, 24

D

DOI 3, 13, 15

F

FTP 8

I

infoseeker 20, 23, 24, 25

L

LCMS 23
 learner 15, 20, 23, 24
 LMS 13, 15, 23
 LOM 3, 8, 9, 10, 21, 23

M

Meta-data
 Version 4
 meta-data 5, 6, 8, 11, 12, 14, 16, 23, 24

O

OpenURL 15, 17, 21

P

PURL 15, 17

Q

query 5, 8, 13, 23, 26, 27

R

roles 20, 24

S

scope 14, 25
 SOAP 3, 6, 7, 12, 18, 22, 27
 standards 21

W

W3C 3, 5, 6, 18, 26, 28

X

XML 4
 XPath 5, 21, 28
 XQuery 3, 5, 7, 13, 21, 26, 28

Z

Z39.50 3, 5, 6, 8, 11, 21

IMS Global Learning Consortium, Inc. ("IMS") is publishing the information contained in this IMS Digital Repositories Interoperability - Core Functions Best Practice Guide ("Specification") for purposes of scientific, experimental, and scholarly collaboration only.

IMS makes no warranty or representation regarding the accuracy or completeness of the Specification.

This material is provided on an "As Is" and "As Available" basis.

The Specification is at all times subject to change and revision without notice.

It is your sole responsibility to evaluate the usefulness, accuracy, and completeness of the Specification as it relates to you.

IMS would appreciate receiving your comments and suggestions.

Please contact IMS through our website at <http://www.imsglobal.org>

Please refer to Document Name: IMS Digital Repositories Interoperability - Core Functions Best Practice Guide

Date: 13 January 2003