IMS Digital Repositories Interoperability - Core Functions XML Binding

Version 1.0 Final Specification

Document Name: IMS Digital Repositories Interoperability - Core Functions XML Binding

Date: 13 January 2003

Table of Contents

1. Introduction

This document constitutes the XML Binding 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 diagram in 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. There are three generalized scenarios considered for the core functions:

1) **Learning Object Repositories.** Searching is performed using the XQuery protocol over XML meta-data, adhering to the IMS Meta-Data Schema. IMS Content Packaging is assumed as the format for Submit/Store.

2) **General Repositories (of resources not purposed specifically for learning).** Assumes use of Z39.50 protocol for searching, with no provision for Submit/Store.

3) **Cross-Domain Search.** Assumes simple keyword searching without internal truncation using the Boolean operators AND, OR, and ANDNOT over a flattened schema of IMS meta-data elements. These elements are listed in Section 3.

Section 4 of the specification describes the use of Simple Object Access Protocol (SOAP) as the underlying messaging service for the core functions supported.

1.1 Nomenclature

ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation-1
BER	Basic Encoding Rules
DC	Dublin Core
DOI	Digital Object Identifier
DRI	Digital Repositories Interoperability
GUID	Globally Unique Identifier
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
LDAP	Lightweight Directory Access Protocol
MIME	Multipurpose Internet Mail Extensions
NISO	National Information Standards Organization
OAI	Open Archive Initiative
REL	Rights Expression Language
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
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, 03c] *IMS Digital Repositories Interoperability - Core Functions Best Practice Guide*, 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. XQuery Search

XQuery Search against full IMS meta-data is the preferred search recommendation for XML-based learning object repositories, thus providing maximum utilization of the rich IMS meta-data for learning objects. XQuery 1.0 and XPath 2.0 are developing specifications of the W3C. XPath is already robust, and the XML Query working group at the W3C forecasts having a stable specification by mid-2002. Several implementations of XML Query are currently available and at least one has been demonstrated against full IMS meta-data.

Detailed information and the actual XQuery specifications can be found at: http://www.w3c.org/XML/Query/. The following statement appears in the W3C document entitled, “XQuery 1.0: An XML Query Language” (http://www.w3.org/TR/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. A query language that uses the structure of XML intelligently can 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. This specification describes a query language called XML Query, which is designed to be broadly applicable across all types of XML data sources.
>
> XML Query Version 1.0 contains XPath Version 2.0 as a subset. Any expression that is syntactically valid and executes successfully in both XPath 2.0 and XML Query 1.0 will return the same result in both languages. Since these languages are so closely related, their grammars and language descriptions are generated from a common source to ensure consistency, and the editors of these specifications work together closely.
>
> It is designed to be a small, easily implementable language in which queries are concise and easily understood. It is also flexible enough to query a broad spectrum of XML information sources, including both databases and documents.
>
> A short example of syntax:
>
> Ordering may be specified at multiple levels of a query result. The following example returns an alphabetic list of publishers. Within each publisher element it returns a list of books, each containing a title and a price, in descending order by price.

```
<publisher_list>
   {for $p in distinct-values(document("bib.xml")//publisher)
    return
      <publisher>
         <name> {$p/text()} </name>
         {for $b in document("bib.xml")//book[publisher = $p]
          return
            <book>
               {$b/title}
               {$b/price}
            </book>
          sortby(price descending)
         }
      </publisher>
    sortby(name)
   }
</publisher_list>
```

The definition of XQuery and its syntax are located on the W3C website. Refer to these documents and future revisions for more complete information:

http://www.w3.org/XML/Query
http://www.w3.org/TR/query-semantics/

3. Z39.50 Search

Z39.50 refers to ANSI/NISO Z39.50-1995, Information Retrieval (Z39.50): Application Service Definition and Protocol Specification, and the matching international standard ISO 23950:1998, Information and documentation — Information retrieval (Z39.50). Ongoing development of Z39.50 occurs through the Z3950 Implementers Group (the ZIG). The U.S. Library of Congress act as the Z39.50 Maintenance Agency. A good starting point for information about Z39.50 is the home page of the Z39.50 Maintenance Agency http://lcweb.loc.gov/z3950/agency.

Z39.50 is a client-server protocol that allows searching of remote databases and retrieving records from those databases in a standard way. It is a session-oriented protocol whose messages are specified using the ASN.1 (Abstract Syntax Notation-1) and encoded using the BER (Basic Encoding Rules) binary syntax. (More information on ASN.1 and BER can be found at http://asn1.elibel.tm.fr/en/standards/.) Encoded Z39.50 requests and responses are transmitted directly over TCP/IP sockets. ASN.1 and BER over TCP/IP are also used for the LDAP (Lightweight Directory Access Protocol) and SNMP (Simple Network Management Protocol). Free and commercial Z39.50 server, client, 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.

Although Z39.50 was developed by the library community to allow searching of bibliographic information and the development of client software that, theoretically, can search any library's catalog, the protocol's extension mechanisms have allowed other communities to take advantage of the features of Z39.50. The definition of bibliographic searching has been extended to include the Dublin Core. Community of interest profiles have been defined for information as diverse as cultural heritage:

- Computer Interchange of Museum Information (CIMI), government and community information: http://www.cimi.org/standards/index.html#THREE
- The Government Information Locator Service (GILS) Profile (http://www.gils.net/), and GeoSpatial Data: http://www.blueangeltech.com/Standards/GeoProfile/geo22.htm

The Z39.50 standard defines both a wire protocol that governs the exchange of messages between the client and the server, and the content semantics that enable interoperable searching and retrieval. The content semantics enable a client to request that the server search a database, specify the search criteria, identify the records that meet the criteria, and retrieve identified records.

Z39.50 does not specify how the user interacts with the client, nor the process by which the server interacts with the database.

The Z39.50 standard defines an extensive set of services, however there are three core services required for search support:

- **Initialization:** At initialization the client contacts the server, a Z39.50 session is established, authentication is handled if required, and negotiations about the session are conducted (e.g., version number, maximum size of records etc.).
- **Search:** The client translates the users search criteria into the standard syntax and submits the query to the target. The target searches the database and reports to the client the number of records in the results set.
- **Present:** The client requests the results of the search by referencing the number of the record in the results set, either individually or in ranges. The client may also specify a preference for the format of the record (e.g., USMarc, Dublin Core). Note the client may process the records for presentation to the user.

Paul Miller (Z39.50 for All - http://www.ariadne.ac.uk/issue21/z3950/) provides this simplified explanation of these core services:

> "Simplifying hugely, Init might be seen as a greeting from the Origin ("Hello, do you speak English?") and a related response from the Target ("Hello. Yes, I do. Let's talk"). Without this positive two-way dialogue, the session cannot proceed.
>
> "A Search request is then transmitted from the Origin ("OK — can I have everything you've got about a place called 'Bath'?"), and is responded to by the Target ("I've got 25 records matching your request, and here are the first five. As you didn't specify anything else, I've sent them to you in MARC format, so I hope that's OK").

> Finally, the Origin uses Present to ask for the data they want ("25, eh? Can I have the first ten, please. Oh, and I don't really like MARC. If you can send me some Dublin Core that would be great, and if not I'll settle for some unstructured text (SUTRS)"), resulting in the transmission of the records themselves from the Target."

Z39.50 Searches are expressed as individual search terms linked by Boolean operators. Each search term has a set of attributes. Attribute sets are developed by communities of interest and are registered with the Maintenance Agency which allocates unique OIDs. The bib-1 attribute set was originally defined by the bibliographic community and has been extended, for example, to enable searching of Dublin Core. A private extension mechanism is also available. The attribute set definition also specifies behaviors for attribute combinations, occurrences, and defaults.

The attributes defined in the bib-1 attribute set are:

- **Use Attributes:** define the access points for a search (e.g., Author, title). The private extension mechanism for Use Attributes has been used to define an experimental set of Use Attributes for IMS Meta-Data (see DRI Best Practice Section 3.5).
- **Relation Attributes:** define how the search term is compared to the values in the database (e.g., less than, greater than, equal to, etc.).
- **Truncation Attributes:** define truncation comparison to the values in the database (e.g., left truncation, right truncation, no truncation).
- **Completeness Attributes:** specify if the search term represents a complete or incomplete field or subfield (e.g., whether additional words can occur in the field or subfield).
- **Position Attributes:** specify where in the field or subfield the search term occurs (first, anywhere).
- **Structure Attributes:** specify the type of search term (e.g., a word, a phrase, a date, etc.).

Interoperability is good in areas of common functionality and gets poorer with distance from core functionality. Barriers to perfect interoperability are due to innate differences in the implementation and functionality of the underlying databases as much as they are due to differences in implementation/interpretation of the standards. Communities interested in improving the available functionality and quality of retrieval may develop profiles which specify a subset of the Z39.50 Protocol. These profiles may be registered as Internationally Registered Profiles. The Bath Profile (http://www.nlc-bnc.ca/bath/bp-current.htm) is one such profile which 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.

Z39.50 is a growing, living standard. At the time of writing, a maintenance version is currently being prepared for ballot. This version incorporates clarifications and amendments, and will fulfill the ANSI 5 year affirmation requirement. ZING (Z39.50-International: Next Generation) "covers a number of initiatives by Z39.50 implementers to make the intellectual/semantic content of Z39.50 more broadly available and to make Z39.50 more attractive to information providers, developers, vendors, and users by lowering the barriers to implementation while preserving the existing intellectual contributions of Z39.50 that have accumulated over nearly 20 years." (from http://www.loc.gov/z3950/agency/zing/zing.html)

Further information about Z39.50, including tutorials, is available from the Z39.50 Maintenance Agency site: http://lcweb.loc.gov/z3950/agency/resources/.

4. Cross-Domain Search

The DRI Project Group is also recommending the development of a minimal search grammar to provide a lowest common denominator search function for cross-domain searches of repositories that may contain very different types of meta-data. This could be based on a sub-set of XQuery grammar. Initial efforts have been made to look at an intersection of XQuery and SRW (an XML-based search tool for accessing Z39.50 repositories). Cross-Domain Searching could be facilitated by an intermediary that could translate a simple query into the appropriate syntax for different search requirements in different repositories.

To simplify the Cross-Domain Search, the lowest common denominator could be Dublin Core (DC) meta-data and search attributes. IMS recommends that digital repositories that are storing learning objects maintain the full IMS meta-data set as defined by the corresponding implementation application profile. Additional search attributes could be provided that allow search clients from other domains to query IMS meta-data repositories. Repositories from other domains could provide additional IMS or DC search attributes to facilitate searching from learning environments.

5. Service Bindings

5.1 General Model

Services interact through the exchange of messages. This section describes the general format of these messages. It also provides details about their binding to SOAP/HTTP as the initial binding message and transport binding chosen.

5.1.1 General Message Structure

The general message structure is shown below.

```
Message
   header
   body
```

The message header is currently empty and reserved for future use. Future uses might include message addressing, message security, message sequence, and so on. The initial choice to specify the message header as empty allows the messaging model to remain consistent with the various emerging standards such as ebXML. In addition, this choice mirrors the SOAP usage that allows messages to have empty headers when using SOAP for a simple messaging model.

The structure of the message body is shown below.

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

The three main elements in the Message Body specify:

- Message Type - the type of message (e.g., ‘Search’).
- Payload - the actual message contents (e.g., an XML document containing an XQuery).
- Audit Elements - zero or more audit elements; usage and structure is currently unspecified.

The element ‘Message Type’ is always required. The presence or absence of the element ‘Payload’ is specified according to message type. ‘Audit’ elements are currently not required.

5.1.2 Message Bindings

The initial message binding specified is XML. An appropriate message schema is to be specified. It is anticipated that this schema will instantiate and refine more general message schemas to be developed by the appropriate IMS Project Groups.

The initial message transport binding is specified as SOAP over HTTP (see below).

5.2 Search with XQuery

Search/Expose allows an infoseeker to issue a query against a repository. The query is expressed as an XQuery request embedded in a message.

5.2.1 Message Structure

The message structure is described below.

```
message
   body
      message type = ‘Search’
      XML Query document
      Audit elements
```

Currently, there is no required element in the message header. The body consists of a required field indicating that this is a message of type ‘Search’, an XQuery document, and zero or more ‘audit’ elements.

The response to the query is also provided in a message. The message structure is identical to that of the Search message:

```
message
   body
      message type = ‘Search Results’
      XML Query document
      Audit elements
```

Currently, there is no required element in the message header. The body consists of a required field indicating that this is a message of type ‘Search Results’, an XQuery document, and zero or more ‘audit’ elements. Any ‘audit’ elements included in the ‘Search’ message must be included as the initial ‘audit’ elements in the ‘Search Results’ message.

5.2.1.1 XQuery Document

The XQuery document is a fully specified XML document fragment, expressed using the proposed W3C XQuery syntax, both for queries and for query results.

Examples of both are reproduced here for convenience.

According to the current XQuery use case descriptions, a typical query document looks like this:

```
<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 a query look like this:

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

5.2.2 XQuery - SOAP Binding

SOAP with multi-part MIME extensions has been selected as the initial binding. This example is based on SOAP Messages with Attachments.

For the ‘Search Message’, the SOAP binding is as follows:

```
<?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 ‘Search Results’, the SOAP binding is as follows:

```
<?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>
```

5.3 Search with Z39.50

Search/Expose allows an infoseeker to issue a query against a repository. The query is expressed as Z39.50 request. Standard Syntax and appropriate bindings are specified in the relevant Z39.50 specifications. Section 3.5 of the Best Practice Guide defines an experimental bib-1 Attribute set Use Attributes extension for IMS meta-data.

5.4 Transport with SOAP

The DRI Project Group is working on additional detail for the SOAP bindings, sufficient enough for implementers to create the appropriate SOAP Services, including relevant XML schemas and examples that use the schemas to illustrate each of the SOAP Services described in this sub-section.

The DRI Project Group recommends the use of SOAP Messages with Attachments for the transmission of IMS defined messages. SOAP is a lightweight protocol for the exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts:

1) An envelope that defines a framework for describing what is in a message and how to process it.

2) A set of encoding rules for expressing instances of application-defined datatypes.

3) A convention for representing remote procedure calls and responses.

Our recommendation utilizes only the first of the three components: the message envelope. The messages being transmitted do not need a data typing mechanism and are too variable to be usable in a remote procedure call environment. A SOAP message is an XML document that consists of a mandatory SOAP envelope, an optional SOAP header, and a mandatory SOAP body. The SOAP message is transmitted in the body of an HTTP message with a content_type of `text/xml`.

5.4.1 Search/Expose - Simple Search

Search/Expose allows an infoseeker to issue a query against a repository. The query is expressed as a Simple Search request. Neither syntax nor bindings are currently specified.

Since Simple Search is based on XQuery with a flattened schema, it is likely that the syntax and bindings will be similar to those specified for XQuery.

5.5 Gather/Expose

Gather/Expose defines a model for aggregation of meta-data by an intermediary. The intermediary then acts as a repository from the point of view of an infoseeker issuing a Search (see Search/Expose above).

The Gather component may interact with repositories in one of two ways. It either actively solicits meta-data (newly created, updated, or deleted) from a repository (pull), or subscribes to a meta-data notification service (newly created, updated, or deleted) provided by the repository or by an adapter external to a repository that enables messaging between the repository and other users (push).

5.5.1 Gather/Expose - Pull (OAI)

“Gather/Expose - Pull” follows the OAI (Open Archive Initiative at http://www.openarchive.org) model as described in the DRI Information Model. That model currently does not specify a SOAP binding. The binding is out of scope for this version of the specification.

5.5.2 Gather/Expose - Push (Alert)

Since “Gather/Expose - Push (Alert)” is specified as a special case of Alert, the binding will be specified in the section addressing binding for Alerts. This is currently out of scope.

5.5.3 Gather/Expose - Push (Adapter)

“Gather/Expose - Push (Adapter)” provides a model where the repository (or an external adapter) forwards the new meta-data to the intermediary in real-time or on a regular schedule.

The push is expressed as an XML document containing the meta-data embedded in a message.

```
Message
   Body
      Message Type = ‘Search Provision’
      XML document
      Audit elements
```

5.5.4 SOAP Binding

SOAP with multi-part MIME extensions has been selected as the initial binding. This example is based on SOAP Messages with Attachments.

For the ‘Search Provision Message’, the SOAP binding is as follows:

```
<?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 Provision</imsmsg:msgType>
      <imsmsg:XML> </imsmsg:XML>
      <imsmsg:AuditElement> </imsmsg:AuditElement>

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

The only type of return message currently supported is the standard SOAP Fault Message.

5.6 Alert/Expose

The DRI Project Group regards the Alert function as a possible component of a digital repository or an intermediary aggregator service and envisions that e-mail/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.

5.7 Request/Deliver

The meta-data about a digital asset should contain a location from which the resource may be obtained. This may be the location for the resource itself, or optionally an intermediary for the resource. For learning objects it is important to address the issues of persistence in these locations and to allow changes or choices of physical location. Location should be expressed as a location (URL), or a method that resolves to a location expressed as a GUID. As URLs cannot be guaranteed persistent, the DRI Best Practice Guide discusses some options for address persistence, see Section 4.4 - Location and Resolution Services and Section 4.8 - Recommendations Regarding GUID Allocation.

About This Document

Title	IMS Digital Repositories Interoperability - Core Functions XML Binding
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 XML Binding for the IMS Digital Repositories Interoperability Specification.
Revision Information	13 January 2003
Document Location	http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_bindv1p0.html

List of Contributors

The following individuals contributed to the development of this specification:

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 Interoperability 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.

Index

IMS Global Learning Consortium, Inc. ("IMS") is publishing the information contained in this IMS Digital Repositories Interoperability - Core Functions XML Binding ("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 XML Binding

Date: 13 January 2003