

Web Services for Digital Libraries

Thomas B. Hickey
OCLC

What are Web Services?

At OCLC Research we take a broad view of what constitutes a Web service. We see any service that uses the HTTP protocol and returns XML as a Web service. There are two main approaches to providing Web services: the Simple Object Access Protocol (SOAP)¹ and Representational State Transfer (REST)².

REST is the simpler of the two. It is a codification of the style of interaction that has formed the basis of the Web over the last decade. Roy Fielding, who was involved in several Web protocol designs, has been able to identify the major features of this style and explain how it interacts to become an identifiable and extendable. REST, whenever possible uses existing mechanisms to communicate, such as HTTP *put* and *get*. The main advantages of this approach to us are the simplicity of REST services and the ease in which they can in turn be used within other services. Since REST services can often be invoked directly from a Web browser, this offers additional possibilities for interaction with services.

SOAP takes a more procedural view of Web services. Functions, their parameters and return values are defined, often using an XML language called Web Services Definition Language³ (WSDL). This enables a very formal and easily portable description of the Web service that makes it fairly easy for a programmer to create a client that will make use of it, or a server to implement the service. The main disadvantage of SOAP is more complication and overhead in the messages compared to REST. This can be a factor in efficiency and ease of integration for those not conversant with SOAP. REST advocates also feel that SOAP hides too much within opaque function calls that would be better exposed using standard HTTP functions whose semantics are well understood.

As a Web service provider we take the rather pragmatic view that we will provide services in the form our users want them, whether REST, SOAP, or some other approach. Our experience is that most of the effort in providing services is the business definition and development of the underlying technologies; multiple protocols should not be difficult to support.

REST

REST takes a very simple approach: requests are sent in as HTTP requests, often as simply a URL. The response is sent back in XML. In some of our services we now include stylesheets to format the XML so that someone coming to the service through a Web browser will see a nicely formatted page rather than the raw XML.

To a great extent, this is how the whole Web works, except that HTML (or now XHTML, the XML variant of HTML) is used in replies.
Here are some digital library examples:

OpenURL⁴:

<http://sfx.bath.ac.uk/sfxmenu?genre=book&isbn=1234-5678>

The Open Archives Initiative Protocol for Metadata Harvesting⁵ (OAI-PMH):

http://alcme.oclc.org/ndltd/servlet/OAIHandler?verb=GetRecord&metadataPrefix=oai_dc&identifier=oai:theses.ulaval.ca:1999/17529

SRU⁶ (a URL protocol for Z39.50 functionality):

http://www.kb.nl/cgi-zoek/srw.pl?query=test&maxRec=10&recSchema=dc_record

As can be seen, the URLs for some of these services can be complicated and contain a fair amount of functionality. We are finding a number of services that can be layered on top of them, especially the OAI-PMH, including name-authority references and registration services. The Open Digital Library (ODL) is another example of a service built using OAI-PMH.⁷

SOAP

SOAP is complex enough that including even one simple SOAP example in its full XML within a paper is difficult. SRW⁸ (Search and Retrieve on the Web) is a good example of a digital library SOAP service, offering the same functionality as SRU, but using SOAP wrappers to move the data. This brings with it a WSDL definition of the service that can ease the implementation of a SRW server within an HTTP server such as Apache. The WSDL definitions and software such as Apache Axis⁹ hide much of the complexity of SOAP from the implementer.

Classes of Web Services

There are a number of classifying digital library Web services. Users and systems need to register different types of objects, search for them, resolve references, navigate hierarchies and other types of links, decompose objects into simpler objects, enhance objects, and transform objects from one format to another.

Registration

Registration services provide persistent identifiers to concepts or objects. They often serve up metadata about metadata. Examples of this include Dublin Core extensions,¹⁰ and the OpenURL formats of digital objects.¹¹ Physical or digital items can also be registered, and cataloging services such as OCLC's offer that sort of registration and control. A similar service is associating identifiers with names agents, such as people and institutions, such as name authority services.

Searching

Search services are all familiar to us. Some sort of search string is typically mapped to a list of records. Searching Web services put this basic functionality in XML wrappers to make it available to higher-level programs the user interacts with. The leading example of this in digital libraries is the Z39.50 work on SRU/SRW. Rather than having to understand BER and ASN.1, programs interact in XML over HTTP. In this case, the main reason not to change over from classic Z39.50 is probably the investment in the current systems. Newly designed systems would find SRU/SRW much easier to implement and use. Slightly outside the library world, systems such as Google are making their services available using SOAP,¹² and finding many different institutions integrating Google in their applications.

Resolution

Resolution services come in many different flavors. Typically, they map from identifiers to data. An example might be a service that accepts an ISBN and returns bibliographic data about the items that ISBN is associated with. Other examples include mapping controlled names to metadata about the people identified, and mapping a subject class number to a textual description of that class. Later in this paper there is an example showing ways this sort of functionality can be delivered by the interaction of PURLs¹³ and OAI-PMH Web services.

Navigation

It is possible to create Web services that facilitate the navigation through a set of links. Of course, the Web itself is an example of general network movement using links, but more specialized services may also be of use. We are working on services that help users move through a classification system such as the Dewey Decimal Classification, or through FRBR's¹⁴ work/expression/manifestation/item entities hierarchy.

Decomposition

There are a number of decomposition services appearing. The University of Southampton is planning on putting several document analysis tools up as Web services as part of the ePrints-UK¹⁵ project to identify references and names within documents. Many times a system has information that can be quite difficult to parse and make sense of. Decomposition services offer the possibility of applying concentrated knowledge and expertise to these problems and making them widely available. These services might be as simple as identifying individual names from a list of names or breaking a class number into its constituent parts, to the more complex problems of exposing the structure of a document or identifying letters/words/sentences within an image.

Enhancement

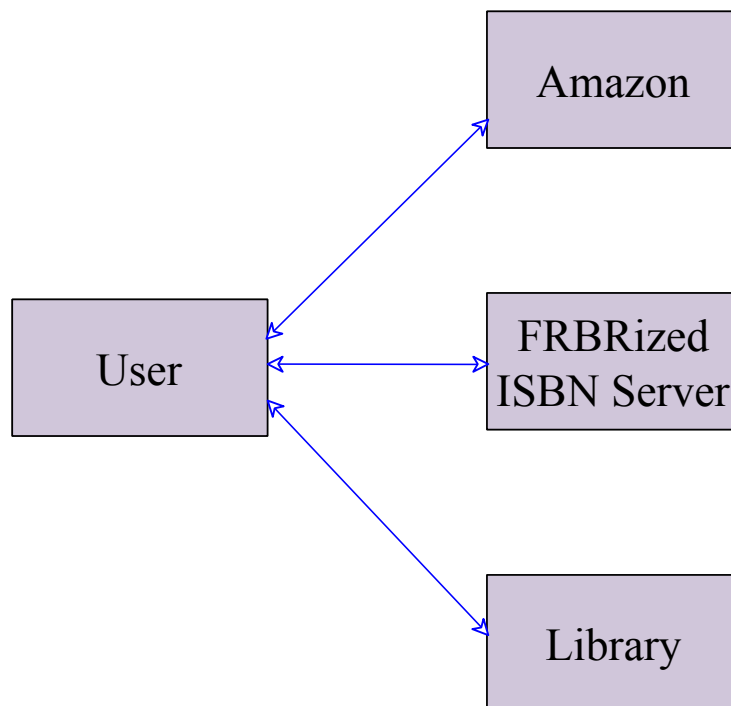
Digital library systems could use any number of enhancement services. An important enhancement are normalization services, such as adding controlled forms of names to metadata containing uncontrolled forms, or adding classification numbers selected on the basis of the text of a document. If links can be identified within metadata records it would also be possible to expand incomplete data. OCLC does that now internally in some of our online services where tables of contents and cover art are added to book catalog records for display. Web services are a way this type of information could be integrated into distributed and disparate services at less cost than maintaining and integrating all the data locally.

Transformation

Many of the above services offer some sort of 'transformation,' so there is overlap between types of services. Transformation services operate at a higher level than many of the other services, e.g. translating the syntax of metadata records from MARC21 to MARC-XML. Somewhat harder to accomplish is the translation of a Dublin Core record (in whatever format) into MARC-21 or between classification systems. At the high-end of these services might be natural-language translations.

Example: Mapping ISBNs to ISBNs using FRBR

IFLA's Functional Requirements for Bibliographic Records (FRBR) lays out a way of relating bibliographic items. We have brought up a service that operates at the work level of FRBR, mapping a given ISBN to all the ISBNs we believe are within that work. A user browsing the Web can then easily invoke this service. For example, the user may have identified a book on Amazon.com and wonder whether the local library holds it:



It operates like this: The user clicks on a button in the 'link' bar of their browser. That button invokes some JavaScript which identifies the ISBN associated with the item displayed, asks the ISBN service for associated ISBNs and then automatically performs the search for items with those ISBNs in the local library's catalog, displaying the results.

So, the user is at Amazon.com
(<http://www.amazon.com/exec/obidos/ASIN/0192816640>):



The JavaScript invoked by the link goes to: <http://purl.org/net/xisbn/0192816640>
Which returns this simple XML file:

```
<?xml version="1.0" encoding="UTF-8" ?>
= <idlist>
  <isbn>0192816640</isbn>
  <isbn>0820312037</isbn>
  <isbn>0820315370</isbn>
  <isbn>0393015920</isbn>
  <isbn>0393952274</isbn>
  <isbn>0393952835</isbn>
  <isbn>0140430210</isbn>
  <isbn>0192811320</isbn>
  <isbn>0192835947</isbn>
  <isbn>0460872885</isbn>
  <isbn>1853262706</isbn>
  <isbn>0874131219</isbn>
</idlist>
```

The ISBNs come back ordered by frequency of holdings within WorldCat, so that for long lists the most commonly held ISBNs can more easily be used in searches.

The result is a search within the library catalog for Humphry Clinker:

The screenshot shows a web browser window titled "CUL Catalog Titles - Microsoft Internet Explorer". The address bar contains the URL: <http://catalog.library.cornell.edu/cgi-bin/Pwebrecon.cgi?BOOL1=any%20of%20these>. The page header features the Cornell University Library Catalog logo and a navigation menu with options like "NEW SEARCH", "RETURN TO TITLES LIST", "SEARCH HISTORY", "PATRON INFO", "REQUESTS", "PREFERENCES", "SAVED SEARCHES", "BOOKBAG", "INTERLIBRARY LOAN", "BORROW DIRECT", "HELP", and "END SESSION".

Database Name: Cornell University Library
 Search Request: Guided Keyword = (0192816640 OR 0820312037 OR 0820315370 OR 0393015920 OR 0393952274 OR 0393952835 OR 0140430210 OR 0192811320 OR 0192835947 OR 0460872885 OR 1853262706 OR 0874131219)[in ISBN]
 Search Results: Displaying 1 through 2 of 2 entries.

Navigation: [previous](#) [next](#)

Sort by: Post Limit

#	ISBN	Title	Date
<input type="checkbox"/> [1]	0820312037 (alk. paper)	Expedition of Humphry Clinker / Tobias Smollett ; introduction and notes by Thomas R. Preston ; the text edited by O M Brack, Jr.	1990
<i>Multiple locations: see record for details</i>			
<input type="checkbox"/> [2]	0393015920	Humphry Clinker : an authoritative text, contemporary responses, criticism / Tobias Smollett ; edited by James L. Thorson.	1983
<i>Library Location: Olin Library Call Number: PR3694.H9 1983 Status: Not Charged</i>			

An OAI-PMH Example

The OpenURL Registry is being built to support the testing of the latest version of the OpenURL standard, which requires that formats referenced by users of the standard be registered. The registry is implemented with a combination of the OAI harvesting protocol and XML style sheets¹⁶ (XSLT) to transform pages into displayable HTML. The registry data itself is held in an OAI repository. That same repository is able to store XML descriptions of the formats being registered. When a Web browser is accessing this information the XML metadata will be rendered for display by applying the XSLT stylesheets associated with the metadata. The site is public at www.openurl.info/registry.



This registration service is actually a slightly extended OAI-PMH repository. The XSLT associated with it makes it displayable in Web browsers, but it is also available as straight XML. In fact, since this is OAI-PMH the whole registry is potentially harvestable for those that would like to maintain a local copy in synchronization with the central copy.

References:

- ¹ W3C. Simple Object Access protocol (SOAP) 1.1. <http://www.w3.org/TR/SOAP>
- ² Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Ph. D. Dissertation, University of California, Irvine. 2000. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- ³ W3C. Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl>
- ⁴ NISO Committee AX. <http://library.caltech.edu/openurl/>

⁵ *The Open Archives Initiative Protocol for Metadata Harvesting.*

<http://www.openarchives.org/OAI/openarchivesprotocol.html>

⁶ SRW – Search/Retrieve for the Web. <http://www.loc.gov/z3950/agency/zing/srw/sru.html>

⁷ Suleman, Hussein and Fox, Edward A. A Framework for Building Open Digital Libraries. D-Lib Magazine, December 2001, v. 7 no. 12.

<http://www.dlib.org/dlib/december01/suleman/12suleman.html>

⁸ SRW – Search/Retrieve Web Service.

<http://www.loc.gov/z3950/agency/zing/srw/background.html>

⁹ Apache Software Foundation. Apache Axis. <http://ws.apache.org/axis/>

¹⁰ The Dublin Core metadata registry. <http://dublincore.org/dcregistry/>

¹¹ NISO Committee AX. The OpenURL Framework Registry. <http://www.openurl.info/registry>.

¹² Google. Google Web APIs – Home. <http://www.google.com/apis/>

¹³ OCLC Online Computer Library Center. Persistent URL Home page. <http://purl.org/>

¹⁴ IFLA Study Group on the Functional Requirements for Bibliographic Records. *Functional Requirements for Bibliographic Records: Final Report*. UBCIM Publications-New Series. Vol. 19, Munchen: K.G.Saur, 1998. <http://www.ifla.org/VII/s13/frbr/frbr.htm>.

¹⁵ ePrints UK project. <http://www.rdn.ac.uk/projects/eprints-uk/>

¹⁶ W3C. XSL Transformations. <http://www.w3.org/TR/xslt>