

# A Case Study of Policy Decisions for Federated Search Across Digital Libraries

Andy Dong  
University of Sydney  
Wilkinson Building (G04)  
Sydney 2008 NSW Australia  
+61 2 9351 4766  
adong@arch.usyd.edu.au

Eric Fixler and Alice Agogino  
University of California, Berkeley  
5136 Etcheverry Hall  
Berkeley, CA 94720-1740  
+1 510 643 1819  
fix@smete.org;  
agogino@me.berkeley.edu

## ABSTRACT

The problem of searching for resources from heterogeneous, networked digital library repositories – sometimes referred to as “cross-collection search” – is becoming increasingly important as the number of on-line libraries available through the Web grows. Federated search is one solution for retrieving information across heterogeneous digital library repositories in real-time with respect to user search requests. While the underlying technologies for federated search are well researched, policy issues concerning intellectual property rights and economic sustainability associated with the selection and application of this technology have not been well articulated. This paper presents an implementation of a federated search service based on SOAP (Simple Object Access Protocol) by the SMETE Digital Library at UC Berkeley. The paper discusses the policy issues associated with the selection of Web services as enabling technologies for federated search and on the establishment of technical specifications in accord with those policies.

## 1. Introduction

Underlying technologies for distributed search across heterogeneous, networked digital libraries are well researched. Client/server-based protocols for searching and retrieving information from remote databases have been widely used since the advent of Z39.50 [3][4] to contemporary efforts such as SDLIP [5]. Essentially two options exist: 1) provide a service such that remote clients can query the repository *synchronously* at the time that an end-user issues the request. Federated search falls into this category; or, 2) provide a service for remote servers to *asynchronously* (with respect to end-users’ search sessions) download metadata in bulk to a local nonvolatile, mass storage device for future searches by end-users accessing that archive. Mirroring, metadata harvesting (e.g., OAI PMH), metadata gathering, and Web crawling are examples of asynchronous techniques.

Policy decisions concerning intellectual property (IP) rights management and sustainability as they relate to the selection of the technical groundwork for creating a cross-collection search service have received comparatively less attention in the digital library literature. These policy dimensions should balance the protection of the intellectual property of the organization while serving to support the organization’s long-term sustainability. [See

Samuelson [7] for a broader review of the law and digital libraries.] This paper presents an implementation of a federated search service based on Web services protocols for the exchange of information in a decentralized, distributed environment. This paper discusses policy issues related to the selection of Web services technologies as building blocks for a distributed federated search by the SMETE Digital Library at UC Berkeley [www.smete.org]. How these policy decisions influenced the technology selections is presented. By contributing to a repertoire of cases of operational digital libraries, the paper seeks to raise awareness of the need to consider the selection of appropriate standards and technologies to enhance the value of the intellectual property of the digital library.

### **1.1 IP and Sustainability Issues**

Choosing a technology approach for a cross-collection search service depends in large part on the type of digital library being operated. Digital resources comprising digital library collections originate from essentially two types of providers:

1. Original Resource Provider (ORP): An ORP hosts the digital resource, e.g., an e-Book, a collection of digital images, a collection of movies, Java Applets, etc. and is the resource. Examples of ORPs include the Alsos digital library for nuclear issues and the Perseus digital library.
2. Metadata Collection (MC): An MC, sometimes called Aggregator, is a collection of collection and item-level metadata resources from one or more ORP and MC. End users search over multiple collection repositories from a central location. The central portal of the National SMETE Digital Library, MERLOT and the SMETE Digital Library are examples of MC's.

A digital library need not be exclusively an ORP or a MC; the digital library may offer both services depending on the scope of the collections. The IP and economic sustainability issues that drive policy considerations will differ, though, depending on how an organization seeks to protect the value and integrity of the resources [1]. Thus, the question is how to select and deploy technology to increase the value of the intellectual property of the digital library. As put forth by Shapiro [8], while rights management is important, intellectual property should be managed to maximize value, not protection. The organization must strategically consider how distribution of the digital library's collection assets enhance the value of the digital library in terms of resource utilization, access (anecdotally known as "eyeballs"), economic sustainability, brand identity, and quality/integrity. Given the stated mission of public digital libraries such as the SMETE Digital Library to make educational resources widely accessible, we must balance accessibility and use of information with fair protection of the IP rights of resource authors, vetting of IP rights, and investments in establishing the digital library. Another related concern is that the organization may want to know which resources are the most popular and who's accessing those resources, which may not be available through indirect distribution mechanisms. At the same time, intellectual property vetting and disclosure agreements between the digital library operator and resource authors must be maintained.

In summary, the IP and sustainability issues the SMETE Digital Library considered included: the cost of distribution of the metadata to increase the number of “eyeballs”; the intellectual property policy of the resources that the SMETE digital library catalogs; IP vetting with other ORP (such as The Math Forum) and MC (such as MERLOT) with which SMETE collaborates; and the risk associated with the loss of the intellectual assets. Being primarily a MC, SMETE opted for the federated search solution with the design goal to support different levels of disclosure from different organizations that may have different IP policies given the uncertain direction of IP policies and business models in this area. The challenge, though, is to “grow the market” for the item-level metadata at the lowest possible cost of distribution of the metadata. Without bulk distribution of the metadata through “mirrors” (i.e., by being an OAI data provider) distribution depends upon the number of clients able to connect to the service and the number of end users who visit the digital library where the collection is housed. The former is not insignificant since, historically, writing and maintaining federated search clients has been a fairly costly enterprise.

## **1.2 Cost Concerns with Federated Search**

The cost of maintaining a cross-collection search service is of concern, where the cost is comprised of the development and maintenance of client and server components. The process of searching over remote digital libraries under the distributed federated search strategy can be decomposed into the following phases.

1. Discovery: Discovery, by the client, of the protocols supported by the server, such as query format, search syntax, and request format.
2. Action: Submitting the request from the client to the server.
3. Response: Parsing the response from the server and displaying the results (if any) to the end-user.

The principle impediment to the adoption of distributed federated search has been the lack of general agreement and adoption of “standards” for request and response protocol and query language. Outside of the (digital) library community, few online services have incorporated Z39.50 [3]. What is needed are technologies that give service providers flexibility in creating a cross-collection search service that suits their specific technical peculiarities while simultaneously lowering the cost to create clients by others to access the service. In the case of federated search, capabilities can be much easier to implement at the client level than comparable harvesting based facilities if the semantics of the search interface and response can be exposed. The federated search client (i.e., the site providing the search facility to the end user) need not be concerned with continuously updating and storing a mirror of the metadata repository at the local site. Instead, one program or component acts as a middleman passing queries to the federated search provider and collecting results in real time. The robustness of the federated search provider and the “cost” associated with end-user time-outs or long delays in response is also to be considered, which can be handled though in the

software development of the client.

SOAP (Simple Object Access Protocol) and WSDL (Web Service Description Language) are emerging Web services specifications developed under the W3C to enable the creation of automated services. The advantages of these technologies have not been overlooked: the ZING project (ez3950) implemented Z39.50 over XML and SOAP. Together SOAP and WSDL support the following key functionality:

1. Gives service providers a mechanism to publish available services, including the semantics and syntax for accessing and consuming the service (WSDL).
2. Allows service consumers the ability to discover services and configure software clients to access remote services.

They comprise a set of technologies to transfer data in an interoperable way, giving service providers and consumers flexible application-to-application messaging. Additionally, they significantly reduce the cost of creating clients due to the availability of open source toolkits such as Apache Axis and The Mind Electric's GLUE that automate the creation of clients. With little development cost, digital libraries can create search services exposing their data, search semantics and response formats, and create cross collection searches with more than one partner. The following table summarizes the policy dimensions the SMETE Digital Library considered and their relation the selection of technology for federated search.

**Table 1. Policy Issues**

Policy Issue		Technology
Maintain IP vetting with resource providers' and individual authors' resources cataloged by the SMETE Digital Library		Federated search
Track popularity and access patterns (e.g., frequent search strings) of resources		Federated search
Ensure quality of organization, structure and presentation of metadata		XML/XSL
Reduce cost of metadata distribution through remote portals		SOAP over HTTP
Enable automated discovery of federated search service		WSDL
Lower the cost of software development for client and server components		Open Source toolkits (e.g., Apache AXIS, Lucene, SOAP::Lite)

## 2. The Architecture

### 2.1 Service Specifications

The SMETE search service support the following input parameters for a search: *key*, *q*, *start*, *maxResults*, and *language*. Definitions for each of the elements are given in Table 2. Clients requiring access to the search service may be given authentication keys ("key") to access the service or for tracking purposes. The key is sent with the rest of the search arguments in the SOAP message. Since SOAP messaging is not necessarily secure, the key is

not secure. This is not necessarily an issue because any abuse of the service may be monitored by the logging mechanism and keys can be revoked and the service denied. (Enforcement is provided through the Validation Layer.) Provisions are included for limiting the number of result elements returned per query, which is both a client side feature (some clients may not be able to handle a large stream of data) and a server-side security feature, to prevent a client query from blocking other requests through long-running search responses. Search continuation and cursoring through any subset of the results are made possible by allowing the service client to enter the starting index for the search. The “language” field specifies the language of the desired records in the response to the query. At this time, only English is supported.

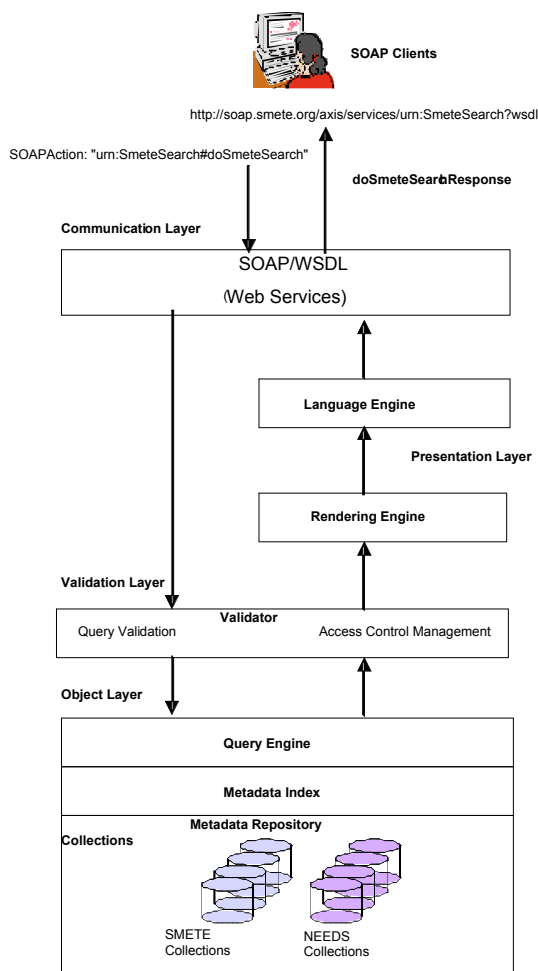
**Table 2. Request Elements Definition**

Element	Definition
<b>key</b>	The <i>key</i> is a unique identifier that may be used for access control management and usage tracking. The key is a string and currently implemented as an <i>id</i> data type in Microsoft SQL Server.
<b>q</b>	This is the query itself. The query is either a string conforming to the Lucene query syntax or an IEEE LOM record in XML.
<b>start</b>	The <i>start</i> field is the one-based index of the first desired result.
<b>maxResults</b>	The <i>maxResults</i> field indicates the number of results to return per response. If the attribute “all” of the tag is set to “true”, e.g., <code>&lt;maxResults all="true"&gt;&lt;/maxResults&gt;</code> , the service will return all records in a single response.
<b>language</b>	The <i>language</i> field is the language of the records in the response. The language identification should follow the XML Language Identification specification.

Two search response formats are supported. SMETE’s search response is syntactically and semantically similar to the OAI-PMH “ListRecords” response type. The main difference is that there are no “responseDate” and “request” fields. Instead, the preamble contains the fields “startIndex”, “endIndex,” “totalResultsCount” and (if appropriate) “error” and the per-record “record” element contains the element “score” to measure the match of the record to the query, scaled from 1 to 100 (perfect match).

## 2.2 Server Implementation

Figure 1 illustrates an end-to-end diagram of the architecture of the SMETE federated search service, from client invocation to server processing and response. The server software was written in Java using the Apache Axis toolkit. The architecture of the federated search service within the context of the digital library infrastructure is designed such that SOAP is the container for the service while various components of the service, such as the query syntax, the per-record metadata format, and the metadata and record language, are interchangeable as needs require. Incoming requests are decoded and converted into a local



**Figure 1 SMETE Search Service Architecture**

essentially a list plus metadata; it contains an entry with the system unique identifier, object type, and score for every result to be returned to the client, and supports result pagination/sessioning, re-sorting, and query serialization so that the same query can be performed again at some point in the future. This is the object that will eventually be passed onto the Rendering Engine in the Presentation Layer, but first it goes through the repository, an object-relational mapping system which uses the type and id information stored for every search result item to load the actual learning object from a content repository (a relational database). The highly abstracted nature of the Query Engine-Metadata Index-Metadata Repository layers makes it easy to construct or configure queries that search multiple indexes and heterogeneous repositories. As long as the items represented by the search index are available as some sort of Java object they can be included in a search results and passed back to the user. This is similar in design philosophy to the search buckets model of ADEPT [2].

When the search results leave the repository, the processing path once again diverges based on the client receiving the data. For each client type, there needs to be a class that

object that contains the query information and any query metadata that may affect how this query is processed by the search engine and data repository. From this point on, until the query results are converted back into a SOAP envelope for return to the client, the query processing pipeline is identical to that used for in-system queries. The query object passes through a Validation Layer where the system calculates whether this query may be satisfied. The query validation layer may also apply additional filters on a query; these filters may be used for a number of purposes, including limiting result counts, and placing certain classes of items off-limits (e.g. based on IP restrictions). Policies for Access Control Management are enforced based on the key element in the request envelope. Usage tracking of specific clients is also possible. Searches are then executed against an index using the Apache Lucene search toolkit. Following the Lucene search, a search results object within the Object Layer is instantiated; this object is

governs the conversion of the search results object into something that the client can understand which is handled in the Presentation Layer. For SOAP clients, the search results, and, most importantly, the embedded objects, are converted into XML, as described elsewhere in this paper. Much of this conversion process is executed using classes generated by Sun's JAXB (Java-XML Binding) toolkit, which is used to define how properties/classes for a particular object are serialized to XML, through the Rendering Engine. This engine converts a native Java object into an XML document given a schema. Subsequently, the XML document could be machine-translated into the requested language in the Language Engine, although this functionality is currently not available. When this Presentation Layer is complete, the contents are passed back to Axis, which places the results into a SOAP envelope for return to the client.

### **2.3 Client Implementation**

Federated search client software is used to connect end users to a federated search service via the SOAP protocol. SOAP client toolkits are also available for most common programming languages, including Java, PERL, C, and AppleScript. The client dispatches separate requests to one or more federated search service providers (user selected) in concurrent, independent threads of execution. The multithreaded implementation of the client queries allows each individual search to be monitored independently, and cancelled if necessary, and prevents the scenario where a failure to respond by one repository prevents the completion of searches against other repositories. Performance of the client is tantamount and the client must be coded fail-safe and with parallelism; else, performance can suffer if there's a problem at the federated search provider. The user of the search client is given real-time feedback describing which results have completed, and which are still pending. This feedback is similar to Web-based airline reservation systems familiar to most regular Web users. At any point in time, the user may interrupt her search and choose to view whatever results are presently available.

When all the individual searches have completed (or the search is interrupted), the list of results is displayed for the user. These results arrive at the search client's system as structured XML data. The federated search operator is expected to provide an XSL style sheet to transform the XML data into the appropriate display format based on the response format specified by the WSDL document. At the SMETE Digital Library, the style sheets transform the results of federated searches into an HTML format identical with the results of local searches such that the results of federated searches appear, to the end-user, indistinguishable from the results of local searches (that is, a search over resources cataloged locally by the SMETE Digital Library).

### **3. Conclusions**

The nature of federated search reveals several policy and technical issues. The most central issue on the policy side is to maintain vetting over intellectual property assets that

maximizes their value not just in monetary terms but in resource utilization and distribution. As the SMETE Digital Library is primarily a collector of metadata, they chose to distribute their metadata through a federated search mechanism based on SOAP/WSDL. Robustness is increased with threading message calls to the services comprising the search. Web services specifications such as SOAP and WSDL allow for interoperability between network services written in different languages and on different platforms. The existence of several open source Web services toolkits lowers the cost of client creation for federated search, thus increasing its chances for wide adoption and adaptation.

Rather than prescribing federated search as the preferred solution for digital libraries, what this paper illustrates is a set of policy decisions that digital libraries might face and how the SMETE Digital Library's technology decisions relate to these policy issues. Given the research interest in finding viable business models for public digital libraries, establishing case studies will provide the empirical evidence of how to manage technology to advance the long-term sustainability of the digital libraries. Managing digital library technologies should include both technology considerations and an organization's internal policy directives such as those described in Table 1. In the case of cross-collection search, digital libraries will likely adopt a range of technologies to suit a spectrum of IP, sustainability, and cost considerations.

#### **4. References**

- [1] Foroughi, A., Albin, M., & Gillard S. (2002) Digital rights management: a delicate balance between protection and accessibility. *Journal of Information Science*, **28**(5), 389-395.
- [2] Hill, L.L., Carver, L., Larsgaard, M., Dolin, R., Smith T.R., Frew, J., & Rae, M.A. (2000, February). Alexandria digital library: User evaluation studies and system design, *Journal of the American Society for Information Science*, **51**(3), 246-259.
- [3] Hinnebusch M. (May, 1997). Z39.50 At Ten Years - How Stands The Standard. *Journal of Academic Librarianship*, **23**(3), 217-221.
- [4] Lynch C. (1997). The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present and Future. *D-Lib Magazine*, **3**(4).
- [5] Paepcke, A., Brandriff, R., Janee, G., Larson, R., Ludaescher, B., Melink, S. & Raghavan, S. (2000). Search Middleware and the Simple Digital Library Interoperability Protocol. *D-Lib Magazine*, **6**(3).
- [6] Paepcke, A., Chang, C. K., Garcia-Molina, H. & Winograd, T. (1998, April). Interoperability for Digital Libraries Worldwide, *Communications of the ACM*, **41**(4), 33-42.
- [7] Samuelson, P. (1998, April). Encoding the law into digital libraries, *Communications of the ACM*, **41**(4), 13-18.
- [8] Shapiro, C. and Varian, H.R. (1998) *Information Rules: A Strategic Guide to the Network Economy*, Harvard Business School Press.

#### **5. Acknowledgements**

The National Science Foundation partially funded this research work under DUE-0085878/DUE-0127580. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.