Unified Approach to Searching across Information Services

Devika P. Madalli
Documentation Research and Training Centre
Indian Statistical Institute, Bangalore, India
devika@drtc.isical.ac.in

A.R.D. Prasad
Documentation Research and Training Centre
Indian Statistical Institute, Bangalore, India
ard@drtc.isical.ac.in

Abstract
Information service providers offer their products through several innovative features. The interfaces are getting more complex with heterogeneous services offered through web pages, journal sites and repositories among several others. Users usually find it difficult to invest time and efforts in learning all different options on these heterogeneous services. In this background, paper presents and demonstrates a unified approach based system that acts as a front-end to several information services. It demonstrates the use of open source tools and toolkits in implementing a unified search and access facility across heterogeneous information services.

Keywords: *Unified search, Z39.50, OAI-PMH, Federated Searching, Metasearching.*

Introduction
Web is popular across domains not only for the variety of resources it offers but as a user-friendly interface that allows seamless access to all types of content including textual, statistical and other multimedia databases. Library and information services and applications have increasingly become web-compliant with libraries, publishers and information service providers offering browser based services. However, while information services are offering innovative features it is often found that end-users have to put considerable efforts in finding their way around different interfaces and facilities.

The present paper discusses search facilities and supporting protocols that are used by various information services. The available technology choices for implementing such protocols are many and generally libraries have opted for structured data based protocols such as z39.50 as against web based services that use the more prevalent harvesters as they have traditionally dealt with highly structured data such as bibliographic databases.

Approaches to Search
There are mainly two approaches to searching that library information services have adopted:

1. Real time/Federated search
2. Non Federated time search

Real time search
In this approach users gives the query and the underlying system performs the search in real time by visiting all the servers, mostly sequentially.

Z39.50 Standard
There are several technologies available to implement real time search, like Z39.50, SRU/SRW etc. These are basically kinds of protocol on which application services can be developed. The 'http' (Hyper Text Transfer Protocol), is the protocol of the WWW that is most widely used for hosting different services including various web transactions and mail. However 'http' is not without drawbacks, when it comes to accessing more than one database using a single interface. For example:

1. HTTP does not support the concept of "session".
2. As it deals with unstructured data, it results in poor indexing and noise in the retrieval.

The Z39.50 standard specifies a client/server-based protocol for searching and retrieving information from remote databases. In other words Z39.50 is a protocol which specifies data structures and interchange rules that allow a client machine (called an "origin" in the standard) to search databases on a server machine (called a "target" in the standard) and retrieve records that are identified as a result of such a search. This specification describes the application service definition and the protocol specifications for real time searching. "Z39.50" refers to the International Standard, ISO 23950: "Information Retrieval (Z39.50): Application Service Definition and Protocol

Specification)", and to ANSI/NISO Z39.50—1995.[1] It is maintained by International Standards and Maintenance Agency, Library of Congress.[2]

Basically, Z39.50 is designed to enable communication by specifying both a general framework for transmitting and managing queries and results, and syntax for formulating queries between computers, typically those containing huge bibliographic data, like library catalogues.[3]

Apprehensions about Z39.50[4]

It is still under development for different extensions as well for basic element mapping

1. Not widely used
2. It is too complex to implement
3. It is often deemed that it is not required any more as we have web
4. It sometimes does not work due to complications in implementations

But,

1. It is a fairly matured standard
2. Fairly widely implemented for LIS work
3. Organizations like museums, art galleries, archives have started using it. Latest version supports non-bibliographic information
4. It is still useful in web environment. In fact, Web provides access to more than one Z39.50 enabled backend databases
5. It promises interoperability across databases
6. Supports maintenance of centralized union catalogues.

As no two databases are expected to be alike with regard to the structure (data elements) and searchable fields, it is required to develop a common abstract model of the target databases. The model should contain the abstract data structure (schema) having the data elements like author, title etc and also the searchable elements as all data elements need not be indexed.

Although Z39.50 is not a database indexing standard, Z39.50 profiles developed for specific communities require a commonly agreed upon database indexing standard. These profiles normally include a minimum set of access points and they should be supported by the database indexes to ensure interoperability[5] between target systems.

Query Protocols:

There are two protocols available for defining the query syntax viz.,[2]

i) **SRU:** (Search/Retrieve via URL)—a standard search protocol for Internet search queries.

ii) **CQL:** (Common Query Language)—standard query syntax for representing queries.

Applications:

There are a number of potential and existing applications of this standard to libraries.[4]

1. Local access to external data sources: The basic search and retrieval functions can be used to extend the number of data sources available for searching at a user workstation. Local and remote databases can be searched using the syntax provided in the local system. This has been the most common implementation of Z39.50 in libraries.
2. Creation of virtual or distributed union catalogues: A group of libraries can use the Search and Present services to enable access from a local origin to many targets. In this way, a user on one library can use the syntax and interface of their local system to search catalogues of other systems in the group. With the ILL Protocol, a group of libraries could provide a virtual union catalogue and mechanisms for resource sharing between them. Issues related to this will be discussed later in the paper.
3. Copy cataloguing using Z39.50: A local Z39.50 origin can search an external database, specify that the records be presented in MARC syntax, and copy them into their local system for inclusion in a local catalogue. This practice is becoming more widespread.
4. Orders for bibliographic outputs: The Extended Services allow a variety of methods to retrieve result sets on a regular basis and have them sent in specified formats. There are a number of possibilities for use of these facilities: SDI services; new and changed records for catalogue purposes; reports for collection development purposes.
5. Updating databases: The Update service of the Extended Services can enable simultaneous updating of more than one target by an origin. This will be taken up further in the discussion of the Union Catalogue Profile.

Software and tools:

There are several software available for libraries to implement Z39.50 standard.

*Z39.50 Gateway Tools***:** Libraries and information providers are adopting the Z39.50 information retrieval standard for accessing their online catalogues. A Z39.50 to Web Gateway allows users to access these databases using browsers such as Netscape. Alternatively, there are software that work as clients and these can be used instead of web browsers. The search operation usually creates a result set, which

is stored on the server and can then be retrieved by the gateway. The features of these gateways include:

1. Querying : The ability of the user to specify and submit queries in a search language.
2. Presentation of results: The results from the searches are displayed to the user.
3. Administration: The setting up and operation of the gateway.
4. Access and resource control: Support for authentication and charging for searches.

Following is the list of some gateway tools:

Name	Platform	URL
Isite	Unix	http://vinca.cnidr.org/software/Isite/Isite.html
Stanford	Unix	http://lindy.stanford.edu/~harold/z3950/www_gateway.html
WebPAC	IBM AIX	http://www.amlibs.com/product/net/webpac.htm
WebCAT	HP, Solaris OSF-1,AIX	http://www.sirsi.com/webcattoc.html
GeoWeb	AIX, SunOS5.2.x,OSF-1	http://www.geac.com/products/library/geoweb.htm

Z39.5 Client Software: The essential function of any Z39.50 client is to allow the user to search Z39.50 compliant databases. The search operation usually creates a result set, which is stored on the server and can then be retrieved by the client. Some of the client software are:

Name	Platform	URL
BookWhere?	Win 3.1, 95	http://www.bookwhere.com/
CanSearch	Win 3.1	http://www.ds.internic.net/z3950/nlc.txt
CIIR's client		ftp://www.usgs.gov/pub/gils/ciir/dtic_a02
DRAFind	Win 95, NT	http://www.dra.com/products/DRAFIND/DRAFIND.HTM
GeoPac	Win3.1,95,NT	http://www.geac.com/products/library/geopac.htm
IrTcl	Unix	http://vinca.cnidr.org/software/Isite/Isite.html
UFO (Fiat lux)	Win 95, NT	http://c134.lib.uci.edu/fiat_lux.htm
Willow	Win 3.1, 95	http://www.washington.edu/willow/
WinPAC	Win 3.1	http://www.als.ameritech.com/winpac.htm
Znavigator	Windows3.1,95	http://www.sbu.ac.uk/litc/caselib/software.html

Limitations

There are some limitations of this type of search:

i) It takes considerable amount of time in visiting each server one by one.

ii) If a server is not working at the time of visit then there will be no results

iii) Does not addresses the User Interface issues that the client finds generally in distributed databases environment.[6]

iv) Does not address the database management issues involved on server side.

Non Federated Time Searching

In this kind of search the query given by the user is given to a server where already harvested metadata is stored and the results are displayed to user as result set. There are some protocols available for harvesting the bibliographic details, viz,

OAI-PMH

Open Archive Initiative for Metadata Harvesting (OAI-PMH)[7], is a web based protocol developed by Open Archive Initiative for harvesting metadata from the repositories who expose their metadata. This harvested metadata from various repositories is further stored to build services for providing search facility. It uses XML (eXtensible Markup Language) over HTTP. The main purpose behind the development of this protocol was to bring application-independent interoperability and extensibility. One of the simplest forms of interoperability among individuals DL systems is the harvesting of metadata.[8]

The current version in existence is version 2.0 which was updated in 2002. This protocol went through many landmarks in its development. OAI-PMH version 1.0 was introduced to the public in January 2001 at a workshop in Washington D.C. and another one in February in Berlin, Germany. Subsequent modifications to the XML standard by the W3C required

making minor modifications to OAI-PMH resulting in version 1.1. The current version, 2.0, was released in June 2002. It contained several technical changes and enhancements and is not backward compatible. [9]

Uses

Commercial search engines have started using OAI-PMH to acquire more resources.[7] Google is using OAI-PMH to harvest information from the National Library of Australia Digital Object Repository. In 2004, Yahoo! acquired content from OAIster (University of Michigan) that was obtained through metadata harvesting with OAI-PMH. Google did accept OAI-PMH as part of their Sitemap Protocol, though decided to stop doing so in 2008. [10]

The OAI's Approach

The OAI-PMH framework explains two classes of participants, viz. Data providers and Service providers.

'Data providers' adopt the OAI's technical framework to expose the metadata about their content. For examples repositories, journal publishers, library catalogues (OPAC) etc. Generally, the exposed metadata is in Qualified Dublin Core but data providers can adopt other forms of XML to expose their metadata. Whereas, 'service providers' harvest metadata from data providers using the OAI-PMH protocol and use the metadata for providing value-added services over them. For example subject gateways, email alerts, consolidation and repackaging services etc.

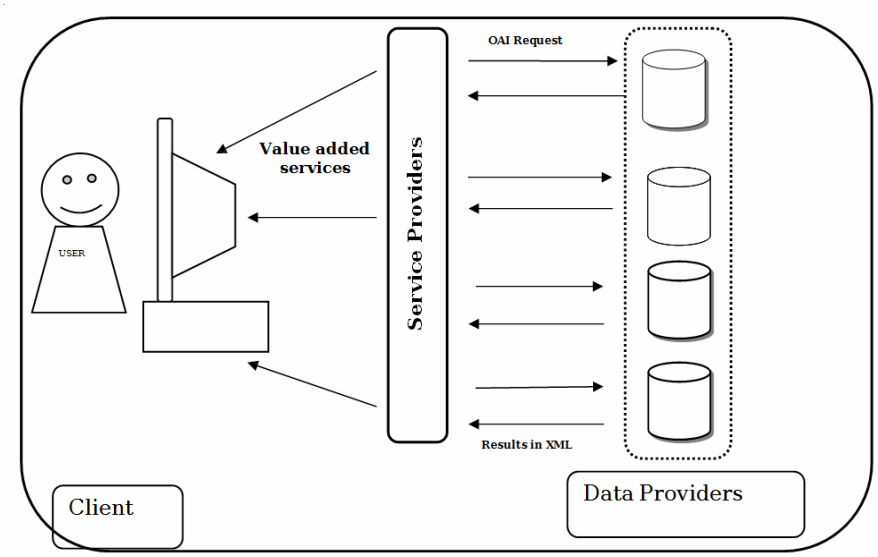

Figure 1: Showing the OAI approach.

The OAI's metadata-harvesting approach might look operationally much different to the Z39.50, but both achieve what's often called "Federated Searching." The federated searches allow users to gather information from multiple related resources through a single interface.[11]

The Z39.50 allows clients to search multiple information servers in a single search interface in "real time", whereas the OAI-PMH allows bulk transfer of metadata from the repositories to the service providers' database. Hence the clients do not need search multiple data providers in real time rather they search the metadata database of the service provider who collect and aggregate the metadata from different data providers.

The technical specifications of OAI-PMH are out of the scope of this paper and are freely available at the OAI website.

Software

A number of software systems are available for data providers for implementing OAI-PMH, including Fedora[12], GNU EPrints[13] from the University of Southampton, Open Journal Systems[14] from the Public Knowledge Project, Desire2Learn[15], DSpace[16] from MIT, HyperJournal[17] from the University of Pisa , Primo, DigiTool, Rosetta and MetaLib from Ex Libris[18], DOOR[19] from the eLab in Lugano, Switzerland.

There are certain software available for harvesting also popularly known as harvesters. One example is PKP Harvester[20] from Simon Fraser University. Another example is MOAI[21], this is developed a company Infrae, the Neitherlands. Apart from regular repositories software it can also be used directly with an SQL database or just a folder of XML files.

These service providers run the harvester which goes to the registered data providers and collect the metadata in XML format. The collected metadata is then parsed to provide integrated search interface. Some of the popular service providers like, OAISTER from University of Michigan Digital Library Production Services, originally funded by Mellon grant[22]. Search digital libraries (SDL), by Documentation Research and Training Centre, ISI, Bangalore, India[23] The OAI-PMH website provides one template where all service providers can register and can get listed[24].

Limitations of Harvesting

a) Regarded as 'blind' protocol

b) Unavailability, unreliability of repository servers

c) Implementation of OAI-PMH v2 incomplete
   - resumptionToken not supported
   - Only ListIdentifiers
   - XML syntax errors

d) Character encoding errors

e) Short lifetime of resumptionToken

f) Expose only a subset of metadata and often no link to full text.

Hybrid Systems

There are at least three main projects on metasearch: LibraryFind[25], dbWiz[26], PazPar2[27]. These are all are released under open source license. LibraryFind project is an open source project of Oregon State University, funded by State Library of Oregon. The dbWiz project is again an open source project for developing a metasearching module from Simon Fraser University with support from the Council of Prairie and Pacific University Libraries(COPPUL) and the Bristish Columbia Electronic Library Network(BC- ELN). The Pazpar2 is a meta search engine from Index Data.[28]

In hybrid systems all the available services of a particular library will be offered through a single interface, where user does not have to visit and search each and every subscribed journal, library OPAC, institutional archives and open access journals through the respective websites or interfaces. A unified approach will provide seamless access to all the information services in very essential in view of the difficulty faced by users in learning about different access mechanism in various interfaces and sites.

We present a hybrid system that is developed by applying two layers; one for searching and another layer for fetching resources out of displayed results. First layer deploys a software compliant with Z39.50 for federated searching and that lists records. The list of records can be further resolved to the actual resources that are heterogeneous such as journal articles, a metadata record or a digital library item. Such resolution is managed in the back-end so that the user experience seamless access to the resources no matter what information service is offering those. The links of the resources is managed through a hybrid system that uses OpenURL link resolver technique. For achieving this, we also deploy a stack of open source software toolkits and modules such as reSearcher software suite[29] developed by Simon Fraser University which consist of:

i. CUFTS

ii. GODOT Link Resolver

iii. dbWiz Federated Search Interface

iv. Cufts Open knowledgebase

CUFTS

CUFTS[30] is an open source serial management software it is used to develop a knowledge base consists of holdings of a particular library, database subscribed journals, open access journals and institutional repositories. This software is written entirely in Perl, it uses PostgreSQL as a database, and Apache as web server.

GODOT link resolver

GODOT[31] (Generalized Online Documents, Ordering, and Texts) is an open source link resolver software, which provides direct access to full text collection using the knowledge base created in CUFTS. It is a unique tool which works as integrated library system, it manages links for both full text as well as printed resources. Even it also works as interlibrary loan request system and provide direct and mediated interlibrary loan requests by the users.

GODOT places a link in the databases, searches the CUFTS knowledgebase for full text, library catalogue for local print holdings, and other library catalogues that are defined for remote holdings.

dbWiz federated Search interface

dbWiz[26] is federated search engine, allows researchers to search multiple databases, websites, catalogues, and other online resources from a single interface, and present the results in an integrated list. It also provides researcher with a facility to choose the desired resources to avoid unnecessary results in result sets.

dbWiz Features:[3]

a. Simultaneous (parallel) searching

b. Searches various database types:
   – Z39.50 targets
   – SOAP targets
   – SQL databases
   – WEB databases

c. Gives users to ability to limit resources searched to defined subset only resources that contain full text articles.

d. Modular design allows addition of drivers to search other kinds of databases

e. Customizable settings:
   – Choose which databases to search
   – Select which web pages to search
   – Select default web search engine
   – Ability to easily organize databases in

categories and to ranking each database ability to easily customize the look and feel of the search interface using a simple template system

CUFTS Open Knowledge Base

Open Knowledge Base[33] is publicly available database of electronic resources right now it contains around 475 resources like ABI/INFORM research: Proquest, ACM Digital Library EBSCO electronic journal service, etc[34].

The libraries using CUFTS are encouraged to participate in contributing titles to an existing CUFTS title list or add their own title lists that are openly shared. This Open knowledge base can be used by libraries to provide accurate information about their collections.

The unified system, IRIS (ISI Repository & Information Services) is implemented at ISI[35] for resources from domains like computer science, spatial Information retrieval, mathematics, statistics and information sciences. This information service consists of

1. ORION Digital Library
2. HORUS Search System and
3. the Federated Search system.

Though, technical discussion of IRIS implementation takes some elaborate explanation, it suffices to state the system has successfully demonstrates ease of access to different information services through a single interface.

Conclusion

Users are often attracted to information systems and services that offer innovative features in browsing and searching collections. But the experience shows that usually the users do not have the time and maybe the patience, to learn complicated features on different information services to extent that they only get used to some services and continue to use only those just to avoid having to learn and memorize how to navigate in different interfaces. The unified approach to different information service presented in this paper simplifies the procedure and provides access to different information in one interface. There are, of course, efficient tools and services offered in the commercial domain. In our efforts we have deployed open source tools and toolkits to offer a low-end and yet robust approach to building an unified searching system most suitable to academic, research and such other information system that involve searching across heterogeneous information services.

References

1. National Information Standards Organization (2003), ANSI/NISO Z39.50 - Information Retrieval: Application Service Definition & Protocol Specification, available at: *www.niso.org/standards/z39-50-2003* (accessed on 25 June 2009).
2. Library of Congress (2007), International Standards and Maintenance Agency, available at: *http://www.loc.gov/z3950/agency/* (accessed on 22rd June, 2009).
3. Lynch, Clifford A. (1991), The Z39.50 information retrieval protocol: an overview and status report, *ACM SIGCOMM Computer Communication Review*, 21(1), 58-70.
4. Prasad, ARD. (2006), A Brief Introduction to Z39.50 Protocol, DRTC presentations, available at : *https://drtc.isibang.ac.in/handle/1849/283*.
5. Lynch, Clifford A. (1997) The Z39.50 Information Retrieval Standard, DLIB Magazine, April 1997, available at: *http://webdoc.sub.gwdg.de/edoc/aw/d-lib/dlib/april97/04lynch.html* (accessed on 25th June, 2009).
6. Payette, Sandra D. and Rieger, Oya Y. (1997). Z39.50: The User's Perspective, DLIB Magazine April, 1997, available at: *http://webdoc.sub.gwdg.de/edoc/aw/d-lib/dlib/april97/cornell/04payette.html* (accessed on 24th June, 2009).
7. Wikipedia contributors (2009), Open Archives Initiative Protocol for Metadata Harvesting, Wikipedia, The Free Encyclopedia, available at: *http://en.wikipedia.org/w/index.php?title=Open_Archives_Initiative_Protocol_for_Metadata_Harvesting&oldid=306393243* (accessed on 23rd June, 2009).
8. Lagoze, Carl (2002). The Open Archives Initiative Protocol for Metadata Harvesting, Protocol version available at: *http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm* (accessed on 23rd June, 2009).
9. Hussein, Suleman; Fox, Edward A. and Madalli, Devika P. (2003). Design and Implementation of Networked Digital Libraries: Best Practices, DRTC Workshop on Digital Libraries: Theory and Practice March 2003 DRTC, Bangalore, DRTC, 2003, available at: *http://hdl.handle.net/1849/29*.
10. Mueller, J. (2008), Retiring support for OAI-PMH in Sitemaps, available at: *http://googlewebmastercentral.blogspot.com/2008/04/retiring-support-for-oai-pmh-in.html* (accessed on 28 June 2009).
11. Amin, Saiful (2003). The Open Archives Initiative Protocol for Metadata Harvesting: An Introduction, DRTC Workshop on Digital Libraries: Theory and Practice March 2003 DRTC, Bangalore, DRTC, 2003, available at: *https://drtc.isibang.ac.in/bitstream/handle/1849/40/H_OAI-PMH_saiful.pdf?sequence=2*.
12. Fedora Commons, Inc. (2009), Fedora Commons Repository Software, available at: *http://www.fedora-commons.org* (accessed on 26 June 2009).
13. University of Southampton (2009), EPrints Digital Repository Software, available at: *www.eprints.org* (accessed on 25 June 2009).
14. Public Knowledge Project (2009), Open Journal Systems, available at: *http://pkp.sfu.ca/ojs* (accessed on 26 June 2009).
15. Desire2Learn Inc. (2009), Desire2Learn, available at: *http://www.desire2learn.com* (accessed on 30 June 2009).

16. The DSpace Foundation (2009), DSpace open source software, available at: *http://www.dspace.org* (accessed on 26 June 2009).
17. HyperJournal (2009), HyperJournal Software, available at: *http://www.hjournal.org/* (accessed on 25 June 2009).
18. Ex Libris (2008), MetaLib: Reach Out and Discover Remote Resources, available at: *http://www.exlibrisgroup.com/* (accessed on 26 June 2009).
19. eLab (2006), DOOR Digital Open Object Repository, available at: *http://door.elearninglab.org/* (accessed on 28 June 2009).
20. Public Knowledge Project (2008), PKP Open Archives Harvester, available at: *http://pkp.sfu.ca/harvester* (accessed on 30 June 2009).
21. MOAI Developers (2008), MOAI, an Open Access Server Platform for Institutional Repositories, available at: *http://moai.infrae.com/index.html* (accessed on 26 June 2009).
22. Digital Library Production Service (2009), OAIster, available at: *http://oaister.umdl.umich.edu/o/oaister/* (accessed on 30 June 2009).
23. Documentation Research and Training Centre (2005), Search Digital Libraries, available at: *http://drtc.isibang.ac.in/sdl/* (accessed on 28 June 2009).
24. Open Archives Initiative (2009), Registered Service Providers, available at: *http://www.openarchives.org/service/listproviders.html* (accessed on 25 June 2009).
25. Oregon State University (2009), LibraryFind Project, available at: *http://libraryfind.org* (accessed on 30 June 2009).
26. Simon Fraser University Library (2005), dbWiz project, available at: *http://dbwiz.lib.sfu.ca/dbwiz* (accessed on 25 June 2009).
27. Index Data (2009), Pazpar2, available at: *http://www.indexdata.com/pazpar2* (accessed on 30 June 2009).
28. Dorman, David (2008). The potential of metasearching as an "open" service, Library High Tech, 26(1), 58-67.
29. Simon Fraser University Library (2009), ReSearcher, available at: *http://researcher.sfu.ca/* (accessed on 25 June 2009).
30. Simon Fraser University Library (2009), CUFTS, available at: *http://cufts2.lib.sfu.ca* (accessed on 25 June 2009).
31. Simon Fraser University Library (2009), GODOT: Open Source Link Resolving, available at: *http://researcher.sfu.ca/godot* (accessed on 25 June 2009).
32. Ehtesham, M. (2007), OpenURL link resolver system, Masters dissertation submitted to Documentation Research and Training Centre, ISI, Bangalore, available at: *https://drtc.isibang.ac.in/handle/1849/414* (accessed on 26 June 2009).
33. Simon Fraser University Library (2009), Open Knowledgebase, available at: *http://researcher.sfu.ca/openkb* (accessed on 25 June 2009).
34. Simon Fraser University Library (2009), CUFTS Targets, available at: *http://researcher.sfu.ca/cufts_targets* (accessed on 25 June 2009).
35. Indian Statistical Institute (2009), ISI Repository & Information Services (IRIS), available at: *http://ir.isical.ac.in/* (accessed on 28 June 2009).