

7-1996


LibSearch: A window-based frontend to remote bibliographic databases on the internet

Ee Peng LIM

Singapore Management University, eplim@smu.edu.sg

Soo-Yin CHENG

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the [Databases and Information Systems Commons](#), and the [Library and Information Science Commons](#)

Citation

LIM, Ee Peng and CHENG, Soo-Yin. LibSearch: A window-based frontend to remote bibliographic databases on the internet. (1996). *International Workshop on the Interfaces to Databases (IDS '96)*. 1-11. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/907

This Conference Paper is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

Ee-Peng Lim

School of Applied Science, Nanyang Technological University
Singapore 639798

Soo-Yin Cheng

Information Technology Institute, National Computer Board
Singapore 0511

Abstract

Over the past several years, a number of wide-area information navigation and discovery tools have been introduced, including WAIS[1], Gopher[2], World-Wide Web[3], etc. In this paper, we describe a graphical query interface to remote bibliographic databases that can be found on the internet. A library query client, called **LibSearch**, has been designed and implemented using a set of APIs based on Z39.50 protocol standard[4]. Z39.50 is an application-layer protocol within the OSI reference model designed to allow library users to remotely access the bibliographic records in the library systems. As increasing number of OPAC systems are being established as Z39.50 servers on the internet, Libsearch provides a user-friendly environment to specify search requests and retrieve bibliographic information from these servers. LibSearch supports: (1) the use of window icons to enable users to easily carry out their search activities, (2) remote accesses to diverse library catalogue systems through a simple and uniform interface, (3) the browsing of search results and storing them in files, and (4) simple user customization.

1 Introduction

The advances in wide-area information sharing and access in recent years have resulted in a variety of public accessible information servers, such as WAIS[1], Gopher[2] and World-Wide Web[3]. By storing information on these servers, information owners provide an easily means for anyone to access the information. Most of these popular servers are set up as independent processes and they support the construction of new and general information sources. For example, a WAIS server is set up by first indexing a number of documents using the WAIS indexer, followed by executing a server daemon which directly evaluates remote access requests to the the indexed documents. Here, the information has been re-indexed or re-organized before the server can effectively operate. Since the server does not depend on any existing search/retrieval engine, no wrapper software is required.

While there is large amount of information stored as plain files in the wide-area network, we must not neglect other kinds of information which have traditionally been managed by legacy information systems which were not designed for internet access. Examples of such legacy systems include database systems, library OPAC (On-line Public Accessible Catalogue) systems, and spreadsheet applications. Due to previous huge investments in software development and personnel training, it is often not viable to abandon existing legacy systems and to re-develop brand new systems to allow internet access. To enable users to access legacy information systems through the internet, it is necessary to include a variety of wrapper softwares supporting different information retrieval protocol standards for different kinds of legacy systems. For example, RDA (Remote Data Access)[5] is a protocol standard designed to allow internet access to databases, and Z39.50[4] is another protocol standard for remote bibliographic information retrievals. The major architectural difference between servers operating on a pre-existing information sources and servers operating on a newly setup information sources is illustrated by Figure 1.

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

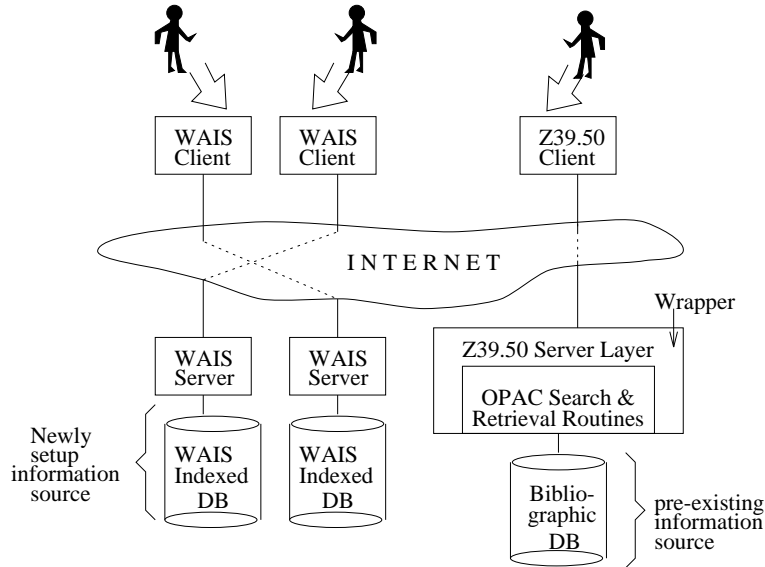


Figure 1: (a) Servers that operate on newly organized data (WAIS) (b) Servers that operate on pre-existing data (Z39.50)

In this paper, we describe a window-based library client known as LibSearch which enables anyone to remotely access bibliographic information from library OPAC systems extended with Z39.50 protocol support. In the last few years, many libraries have automated their bibliographic search services by installing different types of OPAC systems. Provided by different vendors, these OPAC systems support their own user interfaces. Since most users access at most one or two OPAC systems, learning the different user interfaces is still a manageable task. However, as more and more OPAC systems are made available on the internet, conducting searches will be more difficult for someone who needs to access a number of OPAC systems or who needs to access some unfamiliar OPAC systems. In the LibSearch project, our goal is therefore to build a uniform graphical user interface to query these OPAC systems and to make the disparate OPAC user interfaces and local operating systems transparent to the users.

As shown in Figure 2, the design of LibSearch comprises the user-interface layer and the Z39.50 client driver. In the user-interface design of LibSearch, we have considered user-friendliness to be the most important criteria since our ultimate goal is to use it to replace existing OPAC query frontends which are mostly text-based. LibSearch adopts an icon-based graphical interface with each icon representing a specific operation.

Supporting the user-interface operations is the set of APIs (Application Programming Interfaces) provided by the Z39.50 client driver. The APIs support the different search/retrieval services provided by the Z39.50 protocol. Z39.50 is chosen because of its increasing popularity among existing libraries and most OPAC vendors are committed to providing Z39.50 server capabilities over their systems in the near future. Table 1 lists some Z39.50 servers (including the central library at the Nanyang Technological University) which currently operate in the internet. In Libsearch, we have modeled the Z39.50 client-server interaction as an object and the API services as methods of the object. Based on this client APIs, one can in fact develop a variety of user-interfaces for any Z39.50 server.

TABLE 1: Z39.50 EXAMPLES

Organizations	Internet Address	Port	Database Name
Cleveland Public Library	CLEVVXG.CPL.ORG	210	CPL
Data Research Public Library	dranet.dra.com	210	drewdb
Nanyang Tech. Univ. Library	alpha.ntu.ac.sg	210	MARION
AT&T Research Library	research.att.com	210	books

LibSearch has been implemented in the PC Windows environment with WinSocket being used as the TCP/IP layer. The user interface has been implemented using the Microsoft Foundation Class[6]. The first

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

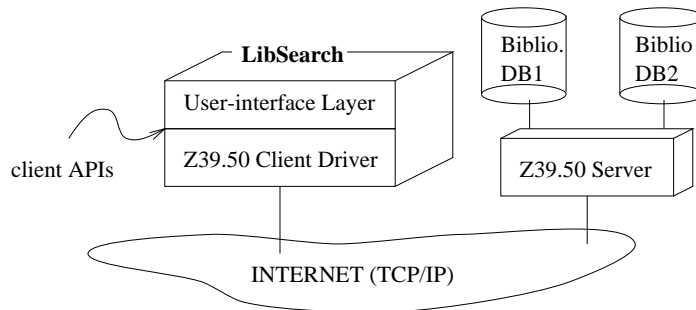


Figure 2: Overall LibSearch Design

version of LibSearch has been successfully deployed to query the various public libraries supporting Z39.50 protocol within the Nanyang Technological University campus as well as the Information Technological Institute, to query the various public libraries supporting Z39.50 protocol. The development and improvement of LibSearch are included as parts of a larger digital library project known as Harp [7].

1.1 Related work

In this section, we summarize some ongoing work in building a user-friendly remote client to pre-existing library OPAC systems. Due to space constraint, we will not attempt to cover all the works, and we will only focus on clients that are designed to communicate with OPAC using the Z39.50 protocol. Readers can refer to [13] for frontends designed to visualize and query remote document servers.

A Z39.50 client called DRA Find has been developed by the DRA organization[8]. Like LibSearch, it runs on in the PC windows environment. On one hand, DRA Find demonstrates some interesting features such as annotating the returned search bibliographic records with appropriate icons representing different record types e.g. books, technical reports, video tapes, etc. On the other, it does not provide value-added services such as storage and retrieval of past search queries and results. Compared to LibSearch, DRA Find does not leave much room for the users to customize the user interface. The searches that can be specified are also limited. Being a propriety software, the DRA Find's APIs to use Z39.50 services are not available for future research prototyping effort.

Willow is an information retrieval tool developed at the University of Washington[9]. It also provides a single, easy-to-use graphical user interface to Z39.50 compliant bibliographic database systems. On the whole, the functionality provided by Willow is similar to that of LibSearch except that the former only operates in the Unix X Window environment. In addition, LibSearch has devised an intuitive and flexible interface for users to specify complex boolean searches involving several ANDs, ORs and NOTs.

1.2 Outline of Paper

The rest of the paper is organized as follows. Section 2 gives an overview of the Z39.50 protocol adopted by LibSearch. The implementation of client driver is described in Section 3. We present the design of LibSearch user interface layer and illustrate the various features using a sample session in Section 4. Section 5 concludes the paper and describes our future work.

2 Overview of the Z39.50 Protocol

To understand the design of LibSearch, one has to first acquire some knowledge about the Z39.50 protocol. Z39.50 is an American National Standard established by the National Information Standard Organization (NISO)[4]. The standard provides a uniform procedure for a information retrieval client to query a remote server supporting an OPAC. It has been later adopted by WAIS to search and retrieve full text documents. Being an application layer protocol in the OSI stack, Z39.50 relies on the ASN.1 encoding and decoding services provided by the presentation layer to format and unformat its service parameters and results.

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

2.1 Z39.50 Services

The Z39.50 protocol defines a number of services to be included by the client driver and server. The core services included in LibSearch are initialization, search, present, and IR- release services. The other Z39.50 services have not been included due to space limitation.

- *Initialization service*: This allows a Z39.50 client to establish connection with a selected Z39.50 server and to negotiate with the server on the parameters used in the subsequent interactions. The parameters to be negotiated include types of information retrieval capabilities (i.e. search, present, etc.), message size, record size, etc..
- *Search service*: This allows a Z39.50 client to send search queries to the connected Z39.50 server. There are a few query types available in Z39.50. However, the standard so far has only mandated that boolean searches (also known as RPN - reverse polish notation queries or type-1 queries) must be supported by all Z39.50 compliant servers. In addition to the query specification, a search request may include the database name(s) to be queried since a Z39.50 server can support multiple bibliographic databases. Depending on the search parameters, one or more bibliographic records may be immediately returned as part of the search response. For each query, a result set is constructed at the server and is identified by a result-set-id.
- *Present service*: Having successfully evaluated a search query against a remote bibliographic database, it is optional for the Z39.50 client to retrieve any subset of the result set using the present service. For a search predicate that qualifies a large set of records, it is advisable to only return a small subset to the client by the search response, leaving the rest to be retrieved by subsequent present requests. In a present request, the client indicates the range of records in a result set (identified by result-set-id) to be retrieved.
- *IR-release service*: This releases the connection with the Z39.50 server and terminates the underlying TCP/IP session.

2.2 Z39.50 Client Driver's Behavior

As shown in Figure 2, Z39.50 protocol addresses the communication between a client driver and a server. Hence, the protocol models the behavior of the client driver and server using different information retrieval protocol machines (IRPMs). In LibSearch, only a number of Z39.50 services are provided. We therefore implemented a client IRPM (see Figure 3) simpler than the original Z39.50 IRPM specification.

As shown in Figure 3, the client driver begins with a CLOSED state waiting for initialization request from the application layer (in our case, the graphical user interface layer). Once the initialization request is received, the driver establishes a new TCP/IP session and sends an initialization APDU (application protocol data unit) to the specified server. Upon receiving the initialization acceptance response from the server, the driver sends a positive initialization confirmation to the application layer and moves to the OPEN state. On the other hand, if an initialization rejection response is received from the server, the driver has to inform the application layer with a negative confirmation, to terminate the connection with the server and to end the TCP/IP session. In the OPEN state, the client driver can process multiple search requests and present requests from the application layer. It is important to ensure that at least one search request is received before any present request. In the OPEN state, the client driver may also receive release request from the application layer. This initiates a release request to terminate the server connection and to end the TCP/IP session.

3 Implementation of Z39.50 Client Driver

In this section, we describe the implementation of our Z39.50 client driver. To enhance the code reusability and maintainability, we have adopted the object-oriented design methodology to model and implement the different tasks of a client driver. As shown in Figure 4, the Z39.50 services are implemented as objects.

The Z39.50 object is the only main object that interacts with the higher user interface layer. Z39.50 services are provided by the corresponding methods of the Z39.50 object. The Z39.50 object, in turn,

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

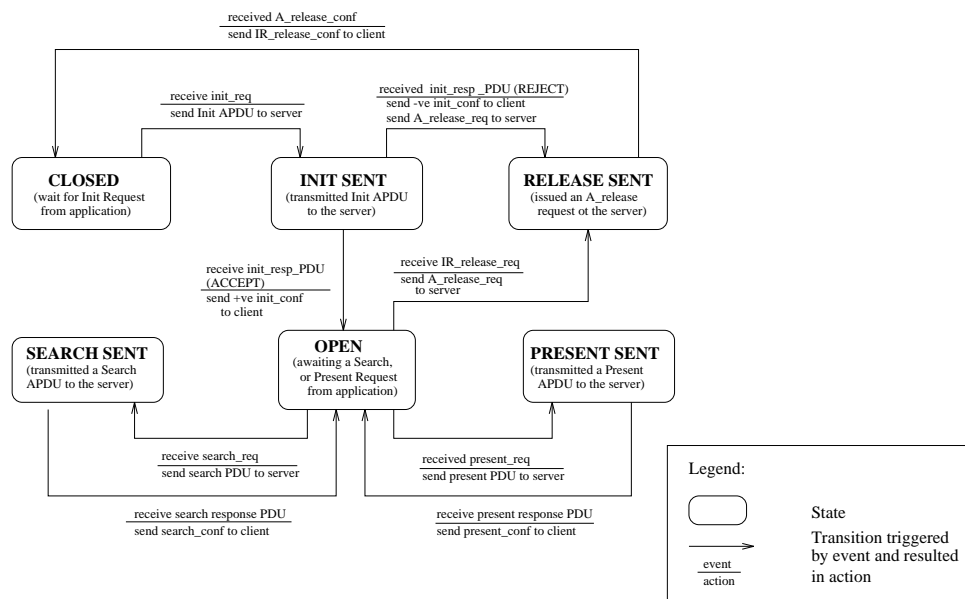


Figure 3: State Machine of Z39.50 Client Driver

instantiates other objects (i.e. socket, initialization, search, present, encode and decode objects) to carry out the specific service requests. In particular, the socket object provides the methods for the Z39.50 object to establish TCP/IP connection to remote server. It also supports sending and receiving data to/from the connected server. The encode and decode objects are created by the Z39.50 object for encoding the outgoing messages to be passed to the socket object and decoding incoming messages from the socket object respectively. Hence, the methods of these two objects are only invoked by the request and response objects created by the respective service objects. For each of initialization, search and present objects, we instantiate the corresponding request and response objects to store the arguments to be sent to the server and to capture the returned parameters from the server respectively.

4 Design of Graphical User Interface

To design the user interface for bibliographic search/retrievals, we have first considered the problems of existing OPAC interfaces as well as the users' opinions. Based on the collected information, we have formulated a useful set of design criteria for LibSearch [10, 11].

To understand the problems of existing OPAC interfaces which operate in text mode, we conducted a small and informal user survey which involved about 1600 students in Nanyang Technological University. The students were reached through electronic mails and each of them was given a fixed set of questions to answer. These are questions related to the OPAC system they used in the campus. About 150 students replied but not all of them completed the questions fully. Their responses have been summarized in Table 2.

TABLE 2: INFORMAL USER SURVEY

Propositions	Votes for Yes	Votes for No	Votes for Neutral
The current OPAC is user friendly	20	13	18
There are problems searching for books/materials	19	12	
OPAC help system is friendly		8	
Library query frontend should be made available in the Microsoft Windows environment	28	8	

Although the above survey is informal, it does indicate that (by looking at the ratio of votes for each proposition) most users feel that the current OPAC interface needs to be improved. Due to the popularity of PCs, having the library query frontend operates in the Microsoft Windows environment is an attractive

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

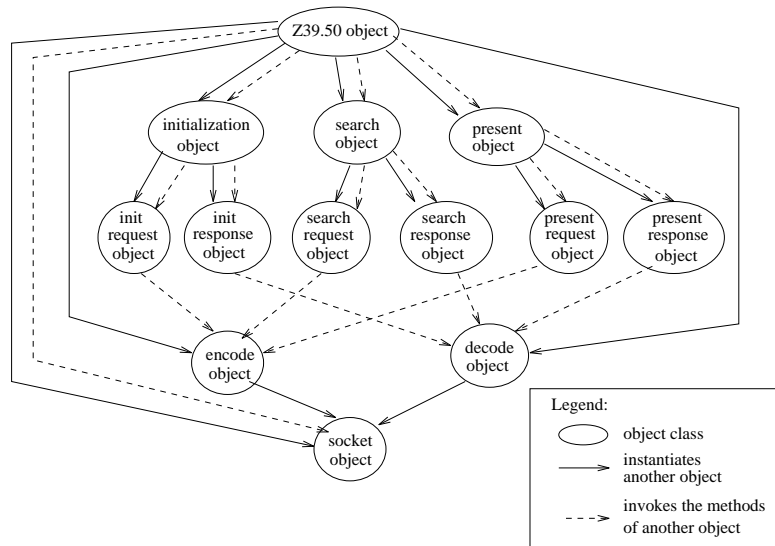


Figure 4: Z39.50 Client Driver Object Design

option. Most users also expressed difficulties search for books/material in different libraries. Lastly, some users feel that the help facility provided by the existing OPAC is not sufficient.

4.1 User Interface Features

In the following, we describe the user interface design criteria adopted by LibSearch. We also present the solution techniques that have been developed based on the criteria.

- **Easy to learn:** An ideal library frontend should be easy to learn. To ensure that users take minimal amount of time to get familiar with the LibSearch user interface, we represent each operations by a unique icon. Buttons, scroll bars, and other window objects are fully utilized to reduce the learning effort. In fact, users can interact with LibSearch using the mouses most of the time.
- **User-friendly help system:** The help facilities of most existing OPAC interfaces are limited. In a window-based library frontend, most search/retrieval operations can easily be performed due to better interface design. Nevertheless, due to a much larger set of operations supported by the frontend, it is still necessary to include a user- friendly help system. In this aspect, LibSearch has incorporated a hyperlink-based help system. By choosing the topics to clarify, users are presented the appropriate help instructions.
- **Reduction of search/retrieval overhead:** While the response time of OPAC interface is usually not a problem in a standalone system, the LibSearch user interface design has to consider the overhead involved in searching remote bibliographic databases. To cope with this overhead, several decisions have been made: (a) Users are allowed to specify their search requests while waiting for the connection with remote servers to be established; and (b) Users are allowed to save their searches and results in files so that they do not need to perform unnecessary searches.
- **Simple user customization:** To cater to different users' need, it is important to allow users to customize the library frontend according to their search requirement. In LibSearch, the following features can be customized: (a) Every user can manipulate the set of bibliographic databases he/she would like to query; (b) The bibliographic fields to be presented in the search results can be customized; (c) By allowing users to save their queries and their results, we essentially allow users to build their own libraries of bibliographic records.

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

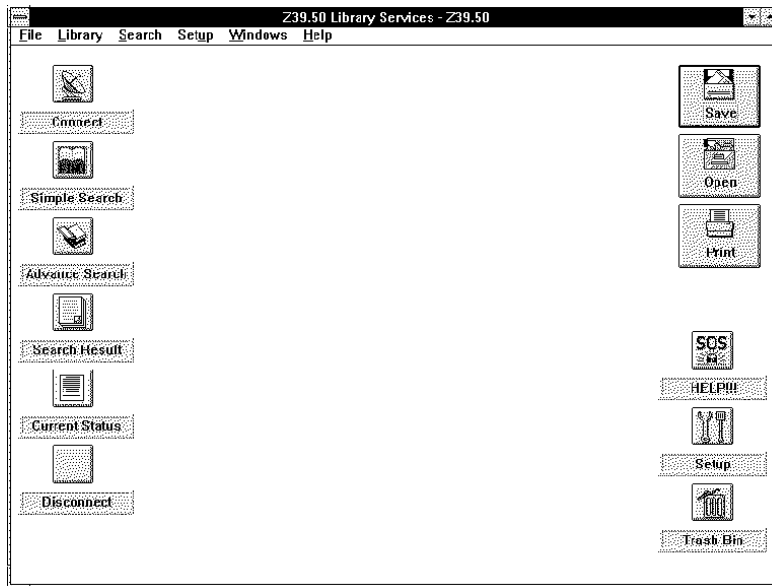


Figure 5: LibSearch First Screen

4.2 Sample Search/Retrieval Session

In this section, we illustrate the essential features of the LibSearch user interface using a sample search/retrieval session. Figure 5 shows the screen that first appears when LibSearch is executed. In addition to the pull-down menu at the top this screen, the main graphical icons can be selected by the user include.

- The connect icon allows one to choose the remote Z39.50 server to connect to.
- The simple search icon is designed for specifying simple bibliographic searches.
- The advanced search icon is designed for specifying complex bibliographic searches. The advanced features include a nested boolean predicate specification and user-defined stored queries. Currently, these features are still being implemented.
- The search result icon is chosen when the user wishes to view the result of his search or the results of past searches.
- The disconnect icon is chosen when the user wants to quit.

When the connect icon is selected, the connection window appears as shown in Figure 6. Here, the Z39.50 server to be connected to can be chosen from a library list. When LibSearch is attempting to establish a connection to the chosen Z39.50 server, the user can open a simple search window by selecting the simple search icon as shown in Figure 7. In other words, the user is free to perform other tasks while waiting for the connection task to be accomplished.

In the simple search window, the user can specify his/her search request by a list of boolean predicates. The "TYPE" box provides a list of attributes (e.g. author, subject, title, etc.) from which the user can choose to formulate the boolean predicates. The boolean predicates formulated are shown in the central clipboard and they can be combined together using the boolean operators AND, OR, and AND NOT. In Figure 7, we specify a query on books written by Gray on the database subject. Having composed the search request, the query can be sent to the server by selecting "SEARCH NOW" button. Once the search result is received, a browseable search result window appears as shown in Figure 8.

In the search result window, a list of bibliographic records returned by the server are shown. When the number of records that qualify the search request is large, LibSearch will retrieve only a subset and display in the window. The user can select "RETRIEVE MORE" or "RETRIEVE ALL" to obtain the rest. To

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

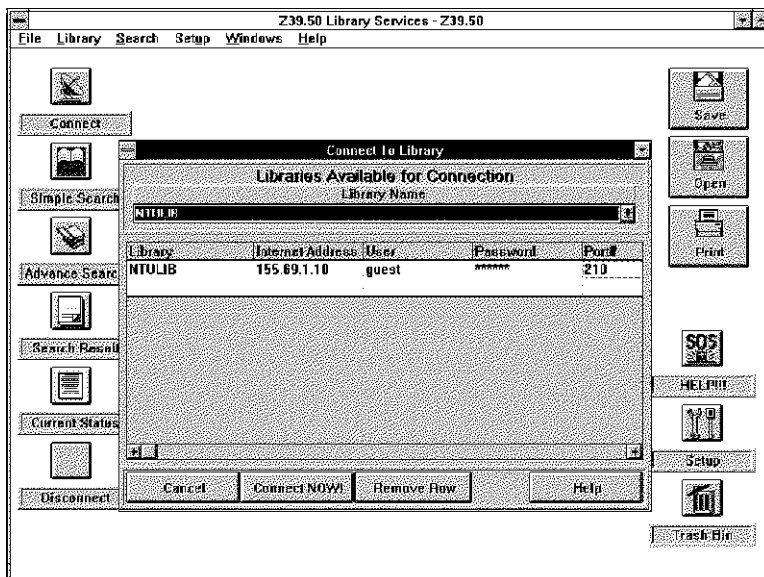


Figure 6: Connect Window

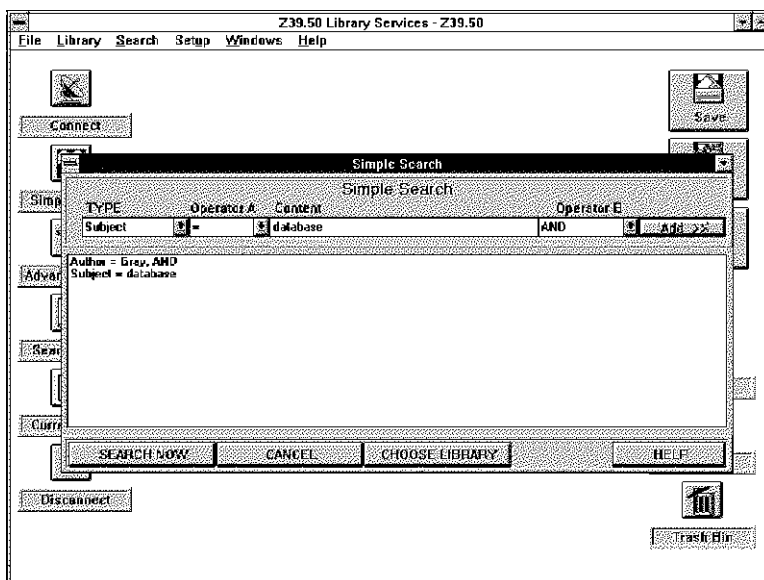


Figure 7: Simple Search Window

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

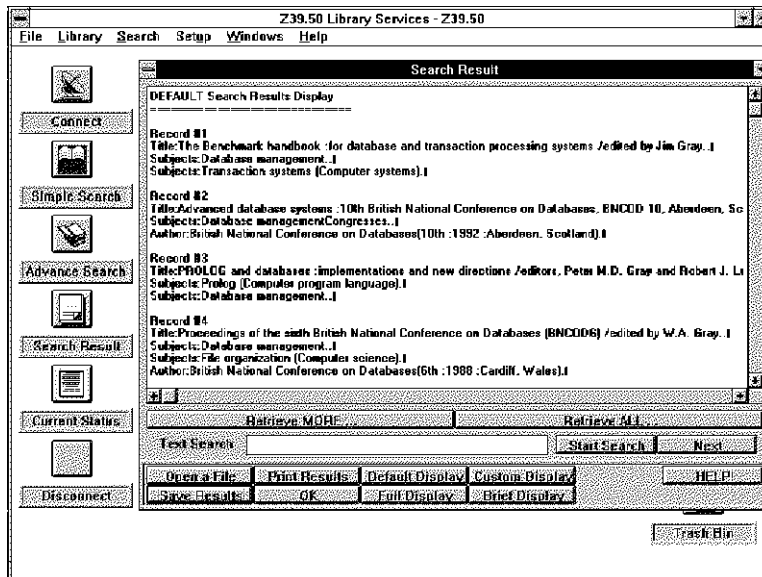


Figure 8: Search Result Window

avoid overloading the user with too much information, LibSearch allows users to select the attributes to be displayed using the "BRIEF DISPLAY" and "CUSTOM DISPLAY" options. The customization of these options can be performed using the setup facility. Unlike ordinary OPAC interfaces, search results can be saved as PC text files by selecting the "SAVE RESULT" button. "OPEN A FILE" is another option which enables users to examine the previously saved results.

LibSearch users can customize the search and retrieval facilities using a setup facility as shown in Figure 9. The items that can be customized include:

- The attributes of bibliographic records to be displayed in the "FULL DISPLAY", "BRIEF DISPLAY", "CUSTOM DISPLAY" and "DEFAULT DISPLAY" modes - This can be easily done by placing crosses(x) on the wanted attributes.
- The Z39.50 parameters that are required for establishing connection with the servers.
- The Z39.50 servers that LibSearch can connect to - new servers can be added by specifying the library names, database names, internet addresses, port numbers, etc. Similarly, servers can be removed from the list.

To further ensure that users are well assisted during the search activities, LibSearch provides a comprehensive help window as shown in Figure 10. The help messages are organized in a document with hyperlinks[12].

5 Conclusions

In this paper, we describe the design and implementation of a user-friendly frontend to remote library OPACs based on the Z39.50 protocol. This prototype frontend, known as LibSearch, consists of a client driver and a user interface layer. The client driver implements the Z39.50 client protocol and encapsulates a suite of APIs within a Z39.50 object which is made available to the user interface layer. As LibSearch is intended for internet users who may want to search bibliographic records using their personal computers, the design of user interface layer has considered several important criteria unique to this context. The useful features incorporated into LibSearch include icon-based menus, storing search results as personal files, a simple setup tool, and a friendly help system. All functionalities mentioned in this paper have been implemented in the

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

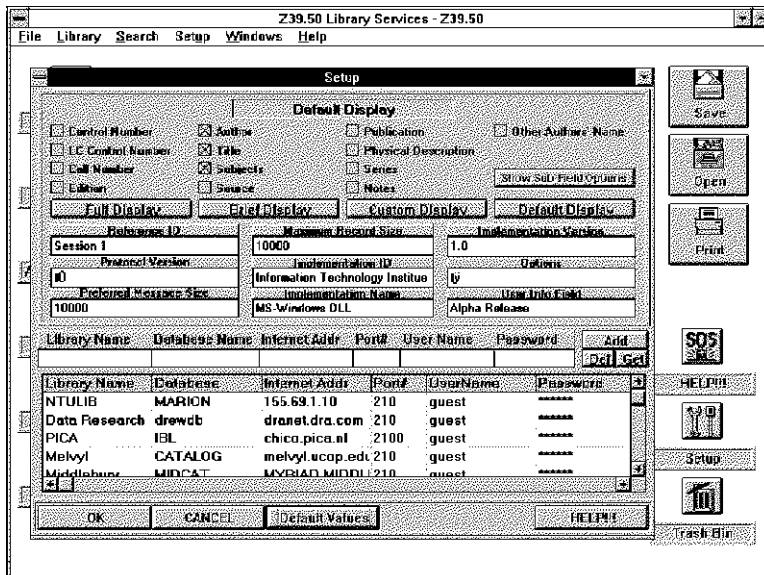


Figure 9: Setup Window

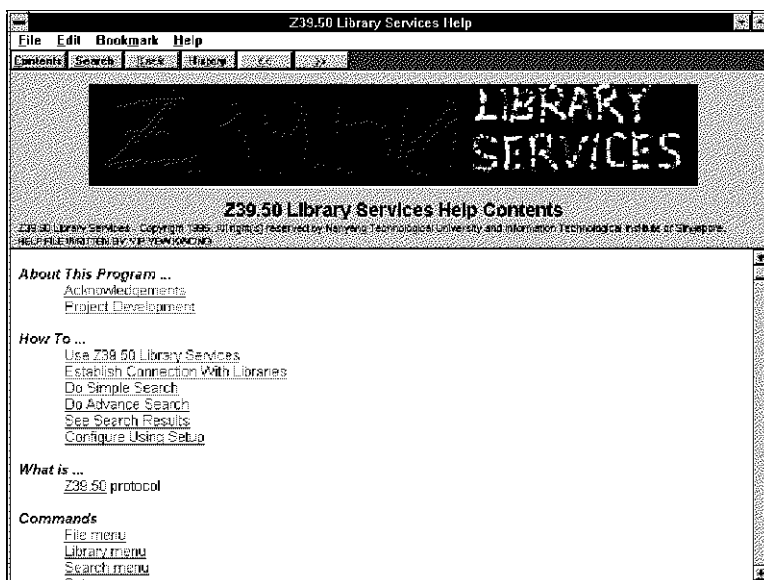


Figure 10: Help Window

LibSearch: A Window-based Frontend to Remote Bibliographic Databases on the Internet

Microsoft Windows environment with the Microsoft Visual C++ and the TRUMPET Winsock library for TCP/IP connections.

LibSearch is just the beginning of our effort to create a digital library environment. Within the Nanyang Technological University and Information Technology Institute, we have initiated an integrated digital library project called **Harp**, which attempts to address information retrieval problems in an internet environment consisting of bibliographic databases, document databases and other user databases. We also plan to experiment the use of LibSearch among the students and staff. The future work that has been planned include:

- Integration of bibliographic searches with document retrievals: This will allow users to perform both activities in a one-stop search/retrieval tool. To integrated bibliographic searches with document retrievals, the association between a bibliographic record and its document content has to be established. We are currently investigating different ways of setting up this association.
- Setting up the user profile by collecting the search patterns of the users. The profile information may contain useful search preferences, such as the classification of servers into categories based on the subjects or keywords so that LibSearch can suggest appropriate servers for future search requests.
- Multiple concurrent searches: At present, LibSearch only allows one search request to be processed at a time. Although users can proceed to formulate another search request before the result of previous request is returned, the new request can only be sent out after the previous result is received. This causes unnecessary delays for the users who would like to send multiple search requests to different servers.

References

- [1] B. Kahle and A. Medlar. An information system for corporate users: Wide Area Information Servers.Connexions - The Interoperability Report, 5(11):2-9,1991.
- [2] M.McCahill. The Internet Gopher Protocol: A Distributed Server Information System. Connexions - The Interoperability Report, 6(7):10-14,1992.
- [3] T. Berners-Lee et al. World-Wide Web: The Information Universe. Electronic Networking:Research, Applications and Policy, 1(2):52-58, 1992.
- [4] ANSI Z39.50: Information Retrieval Service Definition and Protocol, NISO, July 1992.
- [5] ANSI/ISO/IEC 9571-1:1993 & 9579-2:1993 Communication Protocol for Remote Database Access.
- [6] J. Conger, Microsoft Foundation Class Primer. Waite Group, 1993.
- [7] E-P. Lim, B-S. Lee and S-Y. Cheng, HARP: An integrated digital library, Proceeding of International Symposium of Digital Library, Ibaraki, August 1995.
- [8] Data Research Association, DRA Find Installation Manual.
- [9] M. Freedman, WILLOW: technical overview. Technical Report, Computing and Communications Information Systems, April 18, 1995.
- [10] T. Harold, User Interface Design, Addison Wesley, 1990.
- [11] C.M. Brown, Human-Computer Interface Design Guidelines, Alex Publishing, 1988.
- [12] Microsoft. Microsoft Help Compiler Guide, 1993.
- [13] R. Rao etc., Rich Interaction in the Digital Library, Communications of the ACM 38(4), April 1995.