Unit 8 □ Online Public Access Catalogue (OPAC)

Structure

1.0 Objectives

The objectives of the Unit are to :

- Identify basic components of an WEB-OPAC
- Enumerate Functions of WEB-OPAC
- Identify features of OPAC

8.1 Introduction

OPAC is a distinct module of an integrated library management system. It allows searching remotely by various access points, some of which are not possible in manual catalogue. The basic components are :

- User Interface : Allows users to interact with the system. The navigation between "hits" and across retrieved records using hyperlinks is normal in any web OPAC.
- Database : In an integrated library management system, all information about information resources is stored in databases. Authority files may be used for information retrieval.
- IR Options : Information Retrieval options include various searches : author, title, class number, subject, keywords, and location. It also facilitates to limit the search by certain parameters like language, year of publication, and type of information resources required etc.
- Help : Provides guide to user on how to search the database.

8.2 Functions

Remote access to library catalogue is not a new approach. Book catalogues, the forerunners of card catalogues, were an early form of remote access to library information resources. Like all library catalogues, major functions of OPAC may be summarized as follows :

- Locating information resources by known criteria (Author/Title)
- Identify documents on a given subject
- Facilitate resource sharing
- Help in collection development
- Facilitate copy cataloguing
- Facilitate literature search in 24×7 style
- Enable compilation of comprehensive bibliography depending on the need of the user
- Ascertain loan status of an information resource
- Access own account to verify number of information resources borrowed, dates of return and fine due etc.
- Access virtual resources
- Avail enhanced information retrieval facilities (Boolean/Hypertext) and display capabilities.
- Facilitate online reservation of information resources.
- Facilitate online registration for ILL and electronic delivery of digital information resources.
- Act as an interface of online reference service
- Allow user to personalize the interface.
- Enable user to make requests for procurement of information resources online.
- Enable user to check the status of his/her request for procurement(s)
- Facilitate simultaneous search of multiple database

8.3 Subsystems and their functions

The OPAC must provide remote and on-site search facility with an intuitive, easy-to-use Web interface. Desirable features of WEB-OPAC are :

8.3.1 Customization

The Library must be able to create custom web page and/or modify existing OPAC and OPAC Help web pages.

The system must use a single OPAC user login to present all resources and gateways to local databases that an individual is authorized to access. The OPAC must require that a user login to access the library's catalogue, at the Library's discretion. The OPAC module must also support generic or anonymous access to the Library's catalogue.

8.3.2 Searching

1. OPAC must provide keyword, phrase, and Boolean searches, as well as icons linked to Library-assigned search results lists, such as 'Community Organizations', 'Oprah Novels.' 'Holiday Cooking', or 'Science Fair Books'.

2. OPAC keyword searching must be available for every word in every bibliographic record, if desired by the Library. Relevance ranking must be supported. Indexes must be available for searching, including Author, Title, Subject, and Periodical Title indices.

3. The OPAC must enable users to limit (i.e., filter) searches by :

- Publication year (limits retrieval to titles published in, after, or before a specified date, or within a specified date range)
- Language (limits retrieval to titles whose cataloguing information indicates that they were published in the language specified)
- Item type (limits retrieval titles belonging to a specific material type out of the list of possible material types established by the Library, e.g., CDROM, DVD, video, reference book, periodical, etc.)
- Item category (limits retrieval to titles belonging to the user's choice of two specific item categories out of the list of possible categories established by the Library, e.g., Fiction, Nonfiction, Mystery, Children, etc.)
- Format (limits retrieval to titles in a specified broader material type defined by the Library e.g., one of the seven defined MARC formats).
- Location (limits retrieval to titles in a specified permanent shelving location within the Library).
- Library (limits retrieval to items owned by a specified library within a shared catalogue),

4. The OPAC must support broadcast Z39.50 searches of resources and databases.

5. OPAC must display Community Information records in MARC format, if these records are in use by the Library.

6. Community Information records must contain a free text format for data input and support links to scanned images, including map, meeting minutes, etc.

7. The OPAC must enable the user to display maps or graphical shelving plans created by the Library and stored as graphic images.

8. The system must have an icon-based OPAC for children with sets of photo icons, specialized bibliographies such as Suggested Summer Reading, custom pre-set searches designed by the library for standard topics such as science projects or crafts, and preconfigured searches of interest to children developed by library professionals.

8.3.3 User Services/Pesonalization

1. The OPAC must track an individual user's preferences and interests, organized into a list of "favorites" including, but not limited to, authors, subjects, library activities, reading groups, etc. These "favorites" must be included in a user's personal online account.

2. The system must provide user self-service options, or User Services, through the OPAC, including the ability for users to review the status of their accounts and to view custom displays of :

- Bills
- Items charged, with due dates and accrued fines
- Holds requested, with availability status
- Replies to their requests of library staff, with ability to replay, or cancel request
- Notes from library staff

3. OPAC user self-services must also include, at the Library's discretion :

- Renewing eligible items
- Placing holds on items
- Completing online, library-defined Request forms, e.g., ILL, Purchase Request, Suggestions, Self-registration, Reference Questions, etc.

4. OPAC must support Web-based materials bookings/item reservations. Describe your support for material booking by public users.

5. The OPAC module must automatically analyze the Library's overall circulation and display to lists of the Library's most popular titles, authors, or subjects. The OPAC module must update this information automatically.

6. The system must support MARC 856 fields in bibliographic records, so that OPAC users may click on the hyperlink (either actual URL or Library-substituted public note in subfield z of the MARC 856 field) to launch a linked resource, such as a website, digital image or audio file.

7. The Vendor must described all digital media archive modules/products available. The digital media archive module must address the media capture, registration, search, retrieval, and administrative requirements of the Library.

8.3.4 Alternate Language Interfaces

Facilitate change of language of the user interface. Vendor must list alternate language interfaces available for its OPAC.

8.3.5 Content Enhancement/Enrichment

In addition to the standard OPAC, Vendor must offer OPAC content enrichment features that will provide users with images and information similar to online book vendors' sites, such as Amazon.com and barnesandnoble.com.

8.3.6 Broadcast Searching

1. Enable users to conduct simultaneous searches across both Z39.50 and non-Z39.50 targets including :

- Commercial abstract and index databases,
- Library catalogues, and
- Search engines (like Google, Teoma, etc.)

2. The broadcast searching option must return a unified search result for all such searches, regardless of the targets or protocols used to search or retrieve results from each source.

3. The broadcast searching option must :

- Speak to each source in its native language to provide superior results.
- Keep access to resources reliable with continuous updates through subscription-based Resource Plugins
- Deliver results merged into a single interface to enable users to limit searches, sort, and filter.

4. Vendor must describe its proposed broadcast searching option including protocols supported.

8.3.7 Context Management Solution

Vendor must offer a context management solution for collecting and organizing content from many different sources into easy-to-understand and easy-to-use context centers, or 'rooms' for its public users.

The context management solution proposed must be designed and built for use by libraries and library users, with library functions and library information in mind.

The context management solution proposed must provide tools for librarians to build and manage collections of resources in an electronic environment regardless of the origins of the resources.

Vendor's context management solution must enable the Library to organize and present resources including but not limited to :

- Traditional library collections (i.e. our OPAC and anyone else's)
- Digital media
- Online databases and publications
- The "free" Web
- Government and non-profit resources of all kinds

Within each 'room', resources must be presented seamlessly to the user. Content must be arranged according to topic, purpose, and/or audience. The context management solution must enable the Library to purchase completely crafted rooms from Vendor or create its own online context centers or 'rooms'. Vendor must offer all of the following options :

- Carefully chosen content organized into pregrouped collections, and covering a minimum of different subject areas specifically appropriate for libraries and built and maintained by a library subject expert.
- Local customization of 'rooms' based on off-the-shelf "blueprints" supplied by the Vendor
- User-friendly tools to enable original local creation of individual rooms

The context management solution proposed must provide plugins to manage indentification, syntax, and protocol translation with remote resources. The context management solution proposed must provide authentication of resources that require some type of end user identification prior to use, such as a cookie, specific IP address, login, password, or certificate. Online context centers or "rooms" must be delivered through a context server that is responsible for managing user authentication and the navigation between different rooms.

No programming or other technical expertise must be required by Library staff to operate Vendor's context management solution. The context management solution proposed must incorporate seamlessly the Broadcast Searching and Open URL Resolver components specified above. (These components should also be available separately). When a public user enters a context center or 'room' and selects a resource :

- Resource must open in a new window,
- A frame must appear at the top of the new window with a 'return to Library Room X' option,
- The original context room must remains open behind the new window

Vendor must provide sample images of or links to context 'room' arrangements. The context management solution must enable format options within each 'room' :

- Inline frames allowing another web site to be hosted within a 'room'
- Streaming content for displaying syndicated news service feeds.
- XML-style-sheet based content modules to display linked 'rooms' content.
- HTML options allowing for inclusion of external HTML.

8.3.8 Open URL Resolution

Vendor must offer a fully functional Open URL Resolver compliant with current 0.1 Open URL specifications. (Indicate plans for support of future versions, especially the NISO 1.0 Open URL specification). The Open URL resolution feature must accept an incoming (source) Open URL and provide genere-specific resolution services.

Vendor's Open URL Resolver must return and resolve, wherever available, information from :

- Full-text document databases
- Abstract and index databases
- Citation databases
- Content databases with review, tables of contents, first chapters, summaries, author biographies, etc.
- Online library catalogues—both local and remote
- Interlibrary loan and document delivery services
- Web sites
- Electronically accessible resources of all kinds free or licensed, to which the library has access.

Vendor's Open URL Resolver must include a Web based administrative interface enabling the Library to configure and maintain the holdings and subscription data and determine what targets we want to resolve to for any particular information resource and service.

Vendor's Open URL Resolver must permit gathering of statistics thorugh standard webserver-style logs to provide management information on the types of link resolution being done by our users.

8.4 Technology

Initially, most OPACs were developed for LAN. Early LAN OPACs had character based menu driven interfaces. However, with the advancement of the Internet, OPACs first become available via telnet protocol. Today, most of the integrated library management systems contain web based OPAC module. The basic technologies are :

8.4.1 DHTML

Dynamic HTML or DHTML is a technique for creating interactive web sites by using a combination of the static markup language HTML, a client-side scripting language (such as JavaScript), the style definition language Cascading Style Sheets and the Document Object Model.

Some disadvantages of DHTML are that it is difficult to develop and debug due to varying degrees of support among web browsers and that the variety of screen sizes means the end look can only be fine-tuned on a limited number of browsers and screen-size combinations.

8.4.2 Common Gateway Interface (CGI) Programming

A CGI programme is any programme designed to accept and return data that confirms to the CGI specification. The programme could written in any programming language, including C, Perl, Java etc. CGI programmes are the most common way for Web services to interact dynamically with users. The use of CGI programmes on the web server to deliver content dynamically is probably the most used method primarily because this approach is supported on almost all OS.

Major problem with CGI script is that each times a CGI script is executed, a new process is started. For busy websites, this can slow down the server. A more efficient solution is to use the server's API, such as ISAPI and NSAPI. However, Java servlets are becoming increasingly popular.

8.4.3 Dynamic Link Libraries (DLL)

A DLL is a library of executable functions or data that can be used by Windows applications. Typically, a DLL provides one or more particular functions and a programme access the functions by creating either a static or dynamic link to the DLL. A static link remains constant durint programme execution while a dynamic link is created by a programme as needed. DLL files usually end with extensions. dll,.exe,.drv, or .fon.

A DLL can be used by several applications simultaneously. Some DLLs are provided with the Windows operating system and are available for any windows applications. Other DLLs are written for a particular application and are loaded with the particular application.

8.4.4 Java

Java is an object-oriented programming language introduced in 1995 by Sun Microsystems, Inc. Java facilitates the distribution of both data and small applications programmes, called applets, over the Internet. Java applications do not interact directly with a computer's central processing unit (CPU) or operating system and are therefore platform independent, meaning that they can run on any type of personal computer, workstation, or mainframe computer. With Java, software developers can write applications that will run on otherwise incompatible operating systems such as windows, the Macintosh operating system, OS/2, or UNIX.

To use a Java applet on the World Wide Web (www), a user must have a Java-compatible browser, such as Navigator from Netscape Communications Corporation, Internet Explorer from Microsoft Corporation, or HotJava from Sun Microsystems. A browser is a software programme that allows the user to view text, photographs, graphics, illustrations, and animations on the www. Java applets achieve platform independence through the use of a virtual machine, a special programme within the browser software that interprets the byte-code—the code that the applet is written in—for the computer's CPU. The virtual machine is able to translate the platform-independent byte-code into the platform-dependent machine code that a specific computer's CPU understands.

Applications written in Java are usually embedded in Web pages, or documents, and can be run by clicking on them with a mouse. When an applet is run from a Web page, a copy of the application programme is sent to the user's computer over the Internet and stored in the computer's main memory. The advantage of this method is that once an applet has been downloaded, it can be interacted with in real time by the user. This is in contrast to other programming languages used to write Web documents and interactive programmes, in which the document or programme is run from the server computer. The problem with running software from a server is that it generally cannot be run in real time due to limitations in network or modem bandwidth—the amount of data that can be transmitted in a certain amount of time.

8.4.5 Scripting Languages

Scripting languages (commonly called scripting programming languages or script languages) are computer programming languages initially designed for "scripting" the

operations of a computer. Early script languages were often called *batch languages or job control languages.* A script is more usually interpreted than compiled, but not always. The common properties are : they favor rapid development over efficiency of execution; they are often implemented with interpreters rather than compilers; and they are strong at communication with programme components written in other languages. Scripts are typically stored only in their plain text form (a ASCII) and interpreted, or complied each time prior to being invoked.

Types of scripting languages

Application-specific languages

Many large application programmes include an idiomatic scripting language tailored to the needs of the application user. Likewise, many computer game systems use a custom scripting language to express the programmed actions of non-player characters and the game environment. Languages of this sort are designed for a single application and while they may superficially resemble a specific general-purpose language (e.g. QuakeC, modeled after C) they have custom features which distinguish them.

ACS

Action Script

AutoLISP

Text processing languages

The processing of text-based records is one of the oldest uses of scripting languages. Many, such as Unix's awk and, later, Perl, were originally designed to aid system administrators in automating tasks that involved Unix text-based configuration and log files. Perl is a special case—originally intended as a report-generation language, it has grown into a full-fledged applications language in its own right. PHP was originally developed as a specialized language for creating dynamic web content, but is now used by some for general system administration tasks as well.

Awk

Perl

Python

Ruby

PHP

Sed

XSLT

Job control languages and shells

Another class of scripting languages has grown out of the automation of job control-starting and controlling the behaviour of system programmes. Many of these languages' interpreters double as command-line interfaces, such as the Unix shell or the MS-DOS COMMAND.COM. Others, such as AppleScript, add scripting capability to computing environments lacking a command-line interface.

bash

csh

sh

General-purpose dynamic languages

Some languages, such as Perl, have begun as scripting languages but developed into programming languages suitable for broader purposes. Other similar languages-frequently interpreted, memory-managed, dynamic-have been described as "scripting languages" for these similarities, even if they are more commonly used for applications programming.

Cold Fusion

Perl

PHP

Python

Ruby

Smalltalk

Extension/embeddable languages

A small number of languages have been designed for the purpose of replacing application-specific scripting languages, by being embeddable in application programmes. The application programmer (working in C or another systems language) includes "hooks" where the scripting language can control the application. These languages serve the same purpose as application-specific extension languages, but with the advantage of allowing some transfer of skills from application to application.

JavaScript, JScript

Tcl (Tool command language)

References and Further Reading List

1 2005 Basics of library automation standards (http://www.cde.state.co.us/cdelib/technology/atstan.htm). Visited last : 20/.09/2005

2 2005 Request for Proposal for a Client/Server Electronic Labrary System (http://www.libraryhq.com/rfp.doc). Visited last : 12/08/2005.

3 2004 Haravu (LJ). Library automation : design, principles and practice. New Dlehi : Allied Publishers, 2004.

4 2004 Encyclopedia of library and information science. 2nd ed. NY : Marcel Dekker, 2004. 4v

5 1994 Harbour (Robin T). Managing library automation. London : ASLIB 1994

6 1994 Genaway (DC). Integrated online library systems : principles, planning and implementation. NY : GK. Hall. 1984.

7 1983 Rao (I. K. Ravichandra). Library automation (DRTC Refresher seminar 14, DRTC, Bangalore, 1983)

8 1969 Encyclopedia of library and information science. Ed by Allen Kent and Harold Lancour. NY : Marcel Dekker. 1969-. Various volumes

8.5 Exercise

1. What is OPAC.
2. Discuss functions and different subsystems of OPAC.
3. Describe concepts of DLL and CGI.