DRTC Digital Library of Seminar and Conference Proceedings: A Proposed Model

Devika P. Madalli
Documentation Research and Training Centre,
Indian Statistical Institute, Bangalore-59
Email: devika@isibang.ac.in
1 Introduction

The terms ‘Virtual Library’, ‘Online libraries’, ‘Internet repositories’, ‘Digital Libraries (DL)’ and a few more are all used inter-changeably, perhaps with a few subtle differences. There are many claimants from the fields of Computer Networking, Library and Information Science, Communications theory etc., to contributions in the areas of data networks and communications. Whatever the claim, it is the easy availability and feasibility of worldwide communications through the Net that is holding the interest of many researchers, corporate personnel, professionals and academicians alike. Information professionals have realized the new role in the Internet era. The concept behind digital library is not just accessing information on the Internet, though it be may the belief widely accepted. A digital library involves the work of creating and organizing digital collections to disseminating information. The search process may be coordinated just in real time only when the information need arises. Developing such a system is a highly complex process and requires simultaneous advances in many different domains of technical inquiry including user interfaces, search and retrieval techniques, representation of information, and management of intellectual property. It requires combining very large-scale networks with very large-scale file storage and creating digital collections of sufficient depth and breadth to be of compelling interest to working user groups. (1)

2 The background

Documentation Research and Training Centre (DRTC), Indian Statistical Institute, has been hosting seminars and conferences with the objective of sharing the latest knowledge in the areas of both traditional and modern librarianship. The annual seminars are conducted in the latest topics and encourage sharing of scholarly findings, projects and case studies by information professionals. There have also been a series of workshops in the form of refresher workshops. The refresher workshops focus on topics pertaining to Information work and practice. They are handled mostly by the faculty of the Documentation Research and Training Centre and concentrate on highly defined areas aimed at training the working information professionals, researchers and students in information handling tools and techniques.

2.1 Demand

There is a high demand for the seminar volumes from libraries, institutions, individual researchers, teachers and students. Due to various reasons the volumes are not available at times. There are several that are out-of-print also. Some of the volumes are referred to often by universities for exercises such as in curriculum development for professional library science courses. Due to such a demand from professionals, institutions, researches and students in the field, DRTC has undertaken the project of digitization of the seminar volumes to make
1. The seminar volumes conveniently available.
2. For archiving purposes.
3. To provide an efficient retrieval mechanism.
4. To extend it as an online digital library.

2.2 Archiving

It is felt that maintaining the volumes over a period of time may be difficult due the bulk as well as wear and tear. It is essential to take a look at the archiving possibility. Keeping this in mind, DRTC has initiated a project of digitization of the seminar volumes. Of course, the present technology and the resources available are also convenient to digitize the entire collection in-house.

2.2.1 Earlier volumes

Though the efforts are on to preserve the earlier volumes in their original form, it may be difficult with the accumulation of more material every year. Some of the volumes are in cyclostyled forms and others are in photocopied forms. It is to be examined how many of the volumes or pages can be transformed through scanning and text recognition techniques. The rest may have to be transformed manually.

2.2.2 Recent seminars

Interestingly the other problem is diagonally opposite. The recent seminars are in soft copy in some word processor files. They have been quite of volatile nature and have been erased sometimes in upgrades of the disks and sometimes due to lack of space. Yet another problem in the recent years has been regarding compatibility of storage formats. Mostly the word processors such as WordPerfect and Microsoft Word have been used. There has not only been cross package portability problem but also across versions for the same package. The minimum that can be expected is an upward compatibility when versions change and surprisingly even that is lacking.

The discussions so far prepare the ground for the present project. However as an extension of the work, it is proposed to integrate the digitized articles with an adequate search interface and develop the same into an Online DRTC digital library of seminar conference proceedings. The plan of action, the general model of the work involved and the related technologies and techniques are discussed in the sections below.

2 Network and infra-structural support

DRTC, at the Indian Statistical Institute, Bangalore, presently has about 30 systems connected through a LAN. The LAN is connected through servers that work as gateways to Internet using a 64 Kbps leased lines to VSNL.
It has many multimedia systems with scanning facility. On the software side, it has access to the text recognition OCR software (Omnipage Pro) and multimedia components editors such as image editors, video editors, sound recording and editors. It has authoring tools like the Macromedia Director and Authorware packages.

3 Design of the DRTC Digital Library

The work involves various steps from identification of the material for digitization to storing, hosting and retrieving the information. The stages involved in developing the digital library are as follows (Fig:1)

3.1 Scanning

The first step after the initial identification of material to be digitized is the scanning stage. The choice of a scanner is dependent on the type of material that is to be scanned and the software that is being used. In some cases, OCR software is offered along with the scanner.

Broadly, scanners can be classified into the following four types:

- Flat-bed scanners
- Overhead scanners
- Feed-through scanners
- Hand scanners.

The choice of a scanner is dependent on the type of material that is to be scanned and software that is being used. In some cases, OCR software is offered along with the scanner.

A flat-bed scanner or an overhead scanner is ideal to scan bound volumes. In case of flat bed scanner, the input material has to be placed on the glass plate and the scanning head runs over it, whereas with overhead scanner, the scanning head is placed above the input document and the document is kept face upwards.
Fig. 1: Model for Digital Library Development
Feed-through scanners are very much suited for loose sheets. These scanners roll the page over a stationary scanning head. In some cases, Automatic Document Feeders (ADF) are supplied at an additional cost. The advantage of using ADF is that at a time a few pages (around 50 pages) can be fed to the scanners and OCR software takes care of the pagination, even if we have to scan the odd pages first and the even pages next.

Hand scanners force the user to go for multiple passes for an A4 size page. Besides the user should have a steady hand while moving the scanner over the material.

However full-page scanners have become more common on desktops as image input devices for desktop publishing and presentation graphics. Advances in scanning technology, including higher resolution, have increased recognition accuracy.

### 3.2 Text conversion

Although incredible speed has been achieved in the area of computerization, especially in data processing, it is well known that input and output of information is a bottleneck to be reckoned with.

Input is the process of transformation of information into electronic patterns suitable for computer processing. The transformation process involves transforming scanned images into text format. Optical Character Recognition (OCR) is one such technology, which is gaining much importance at the data input stage.

OCR can be defined as "a high speed process of recognizing and translating machine printed or hand printed words, letters, symbols and numbers into computer processible information. The data is directly machine readable while still being readable by people".

OCR technology converts printed characters into electronic ones that can be processed by a computer. Image scanners translate graphics (e.g., line art, photographs) into digitized images for computer processing. Both OCR and image scanners use a similar technique to convert their data. Software within the computer determines how the data is to be manipulated and eventually stored.

Concurrent with developments in microprocessor technology, OCR technology made dramatic strides in the late 1970s and early 1980s. The present generation of OCR scanners can read several type styles commonly used in offices (e.g., Courier 10, Pica, Prestige Elite, Gothic etc.). Scanning accuracy also has improved and availability of new software adds a number of processing steps to improve accuracy further.
**OCR system**

OCR comprises of two steps in translating characters on a page into a digitized form

i. Optical Scanning

ii. Recognition System

The optical scanning system is used to scan a page and store the resultant image in a computer, whereas the recognition system tries to analyze each character and translate it into machine language. Usually, the scanned image of a page is stored in a graphic mode with which no editing of information is possible as the stored image is not textual information to the computer. It is the function of the recognition systems to transform the stored image into textual information. In most computers all the alphanumeric characters are assigned ASCII (American Standard Code for Information Interchange) value. For example, the ASCII value for the character ‘A’ is 65 (decimal) and that of ‘B’ is 66 and so on. When a page is scanned, the scanner generates an image or picture of the character ‘A’ in the same size as it appears on the page. The task of the recognition system is to identify the character ‘A’ and assign the respective ASCII value. In other words, the OCR system converts the graphics file into an ASCII file. Then only it is possible to process or edit information that is read using a scanner. Normally, when data is entered a computer generates a "text file". Such a file could be manipulated using any text handling software like word processors, spreadsheets, database management systems (DBMS), etc. OCR systems today directly allow a file to be saved in all popular word processor formats. Advanced features includes saving the scanned data as spreadsheets also.

**Scanning methods**

The current OCRs are primarily desktop units that contain a page transport mechanism (single-sheet or automatic feed), a scanning unit and some micro processor control to store and recognize characters. Most OCRs have some operator controls for contrast, brightness and page feeding. Increasingly, scanner manufacturers have passed this kind of control to the operating software, particularly for graphics.

Before a character can be converted into a digitized representation, it must pass through two systems: an optical scanning system and recognition system (the analysis and decoding of optical output). The optical system not only governs speed and flexibility of the scanner, but also determines the range of inks and paper surface
qualities needed. The scanning process determines the presence or absence of mark or stroke by the amount of light reflected from the area being scanned.

With text, scanning is only the first step. The OCR hardware converts the analog signal pattern from each pixel and digitizes it into a matrix of binary data. This data table, stored in RAM, is then checked against a table of characters stored in programmable read-only memory (PROM). The OCR scanner compares the data against its set of characters and converts successful character matches into ASCII format. There are two basic methods of character recognition in OCR.

i. Matrix or template matching
ii. Feature extraction, also known as pattern recognition

Matrix matching

It is used to recognize mono-spaced characters, matrix matching compares scanned data against a standard character template. Using this method, a bit map of characters is examined through predetermined hot zones, rather than each character bit. Digital templates for various typefaces, like Courier and Prestige, are stored in PROM on the OCR, and can be changed depending on the typeface of the material being scanned. Scanned characters can be run through the table in a loop several times until a successful match is found. The process of matching characters within the loop is accelerated by interposing the most common letter over and over in the table (e.g., letters like e, s, t). If a match cannot be found within the loop, the OCR tries tilting the characters and running the loop again. Some systems run characters through a digital filter to clean up the copy and try matching again. When every other method fails, the operator will be alerted and asked to enter the character, or an unknown character flag (usually a ? or ~) is sent.

Feature extraction

Feature extraction, also referred to as pattern recognition, goes beyond Matrix matching. Instead of looking for a exact match against a template master, the feature extraction process scans the shape of characters and compares them against shape tables by examining the arrangement of bits (rather than the bit map) composing the letter. Feature extraction technique enables the scanner to scan a vertical bar followed by a circle connected halfway down, and based on this information, to form a pattern recognition algorithm to determine that the bar and circle describe the letter "b". Conversely, scanning a circle followed by a bar describes the letter "d". Feature extraction offers much more flexibility than matrix matching because the shape tables can be taught to read a variety of type styles.

Feature extraction, unlike matrix matching, is able to scan documents that are proportionally spaced, while offering the ability for the system to learn a typeface, including italic and boldface characters. An operator starts the process by feeding the proper font information to the shape table and, in most cases by running a series of
scans to instruct the machines as to specific patterns unique to that face. The algorithm used to describe these characters requires considerable computing resources and exceptional resolution on the part of the scanner. Consequently, feature extraction, which is based on topographical character analysis, examines more bits than does matrix matching and therefore often takes longer to make a match. Because of the more precise matching procedure, however, a wider range of fonts can be read.

A few years ago, most OCR systems on the market could not read dot matrix print, and few, if any, could read typeset documents. Today, many OCR systems are able to read dot matrix print, and several vendors offer scanners that can read almost any font or format (omnifont) including typeset material. The trend toward omnifont readability vastly broadens the range of document sources from typewriters and daisy wheel printers to laser printer, offset presses, typesetters, photocopiers, and letter quality, dot matrix printers. Today, the OCR software has the option to train the recognition unit to recognize the strokes in manuscripts.

**Evaluation of OCR software**

Optical character readers are devices that scan printed information and convert it to digital form for storage or processing in an editor or word processor. In the past, OCR devices were used primarily in industrial and retail environments for inventory control and data entry. Earlier machines were large, expensive, limited in function, and frequently inaccurate. Typestyle recognition was restricted to one or two highly specialized fonts, and special paper was required. Today's scanners have evolved into faster, less costly, more fully featured and accurate units that are available in a variety sizes. A large majority of the models are desktop units, which often combine both OCR and image scanning functions and a growing number of hand-held readers.

Prospective buyers of OCR readers should examine their existing office equipment (including typewriters, word processors, and computers) and determine their scanning applications in order to choose the most efficient scanner for their need. A text-processing scanner should accommodate the most common type styles, document formats, and paper types used in current applications. Scanning speed and compatibility with existing office automation equipment must be considered as well.

Following are some parameters which are to be considered in evaluating the OCR system (includes both scanning and recognition system)

1. **Manufacturer and model**

   The manufacturer is the company responsible for labeling and packaging the scanning units and selling the unit to the end users. However, a separate, unspecified company may manufacture the actual scanning unit. Manufacturers often sell their equipment through dedicated sale outlets, distributors, or retail dealers. In any case, the reputation and track record of the manufacturer and vendor is important. Also the technical support statements and warranty period offers have to be compared when there are many contenders.
ii. **Type**
This parameter is used primarily to check for text processing data entry, and whether the unit is a desktop model, hand unit, or floor-standing console.

iii. **Applications**
The types of applications that the unit supports are considered in this parameter. Applications include desktop publishing, word processing, spreadsheets and archiving functions, among others.

iv. **Host system**
The host system that the OCR scanner needs includes components such as PCs, mini-computer, mainframe, and workstation.

v. **Power requirements**
The power requirements of the unit.

vi. **Document handling**
Delineates document standards and paper handling capabilities. Size of the document (Width x Length) – identifies the minimum and the maximum document width and length, that can be accommodated by the unit.

*Feed technique:* This denotes the type of feed mechanism that is used on the scanner. The two most commonly used types of feeder are friction and vacuum. There are a number of hand-fed units also available.

*Document feeder capacity:* The maximum number of sheets that can be handled by the unit's automatic document feeder.

*Ribbon requirement:* The type of ribbon needed for use with the system, if there is a specific requirement. Most manufacturers recommend print that has been generated by single-strike carbon ribbon, although the standards have become more flexible with recent systems.

*Paper weights:* The paperweights that can be accepted by the unit's feeder and transport mechanism.

*Mixed paperweights:* The ability of the document feeder to accept paper of mixed weights in a single stack.

vii. **Recognition**
These are the acceptable parameters of the scanned documents. It is important to note that the text that does not conform to recommended specifications will increase the number of scanning errors.
Character recognition technique: Indicates the method by which characters are recognized, such as matrix matching, feature extraction, or a proprietary method, including those that are based on artificial intelligence often called Intelligent Character Recognition.

Scanning technique: The method for optically converting the printed images to electrical signals. The most commonly used technique is CCD array.

Print source: The source materials recommended for scanning (e.g., typed, machine-printed, hand-printed, clear photocopies, dot matrix, typeset, photographs, line art). Scanning accuracy is highly dependent upon the quality of copy used.

Character set: That is upper and lowercase letters, alphabetic and numeric characters, and foreign character sets.

Lines per inch: The maximum and, when applicable, the optimal number of lines of text per inch that can be determined by the number of times the scanner stops both horizontally and vertically on a page. In addition, it refers to the maximum resolution (in dots per inch) that is supported by a scanning unit.

Line spacing: The ability of the scanner to read single, single and-a-half, double, or triple line spacing.

Number of fonts resident: The maximum number of standard and optional type styles that are recognized by the unit.

Fonts recognized: Type styles recognized by the system, OCR-A and OCR-B were designed specifically for OCR applications but are not widely used in today's text processing market, fonts such as Prestige Elite, Letter Gothic, Pica, Elite, and 10 and 12 pitch Courier, have more widespread OCR use, and a growing number of units are able to read dot matrix print and/or typeset material. Units that are designed for both text and data entry often have the ability to read specialized numeric fonts and hand printed data.

Pitch: The number of characters per inch that can be read by the unit, with 10 and 12 pitch being the most common. Many modern units are able to read proportional spacing as well.

Auto font/auto pitch ID: Whether font recognition and document pitch can be handled automatically by the scanner.

viii. Performance

This refers to the speed and accuracy level of the scanning device. Performance is dependent upon the quality of the input copy as well as the capabilities of the equipment.
Scanning speed: The optimal scanning speed of the unit, generally expressed in characters per second (cps) or inches per second (ips).

Full-page scan time: The approximate amount of time, in seconds, that it takes to scan a full page. The actual number of pages per hour will depend upon the amount of text and/or type of image contained on the page as well as a number of other variables including clarity of copy.

Resolution: Indicated in dots per inch (dpi), of the unit.

User-selectable resolution: The option of whether resolution can be selected by the user. The higher the selected resolution, the slower the scanning speed.

ix. Character accuracy ratio

The accuracy ratio with regard to character substitutions and rejections based on the number of scanned characters.

Substitutions/rejections: The ratio of character substitutions and rejections as compared to the number of characters scanned. Character substitutions are used only in word processing scanner, since a higher level of accuracy is required for data processing applications.

x. Error control

The means by which the scanner is able to perform error control.

Flags character to output: Whether the scanner is able to flag the existence of an unrecognizable character through the use of a special character or symbol on the output device.

Marks error on document: Whether the unit can mark the document to indicate the existence of an unreadable character.

On-line correction: Whether the scanner offers on-line correction. In conjunction with display capabilities, this feature allows the operator to correct errors before they are transmitted to the word processor or computer. The display enables the operator to view an image of the unreadable character, or the readable characters with a blank space or an alternate character in place of the unreadable one.

Type of error display: The type of display that is used for error control. The most commonly used displays are Liquid Crystal Display (LCD) and CRT.

xi. Scanning control

The methods of scanning control offered by the unit.
Brightness/contrast: Whether the unit provides control over the amount of light reflected from the image and whether upper and lower limits can be set on the reflected light.

Data compression: Whether the unit is capable of compressing black and white areas through algorithms that encode portions of the black and white areas.

Mixed mode: Whether the unit is capable of reading more than one input format simultaneously (Such as text and image).

xii. Editing
Editing and programming capabilities offered by the unit.

Text display/size: Whether a text display, which allows the operator to view varying amounts of text (ranging from several characters to an entire page) during the scanning process is available.

Deletion symbol editing: The unit to interpret a single unique symbol or sequence of symbols that are entered by the typist of the original draft.

Programming capabilities: Whether the system can be programmed for special applications and whether this is accomplished by vendor customization, header sheets, diskettes, tape wafers, or other methods.

Auto rescan: Whether the ability to rescan the document automatically.

xiii. Output/interfaces
Communications protocols and device compatibility

Output media: The ability of the system to output to such media as magnetic tapes or diskettes.

Standard interfaces: The type of interfaces available with the unit (e.g. RS-232C, bus, typesetter).

Protocols/formats: Communications protocols and formats used for data, text and graphics transmission (e.g., async, bisync, ASCII, EBCDIC, SCSI, TIFF).

xiv. Sales and services
The initial delivery data of the system, the number of installations, and the sales and services channels available. In addition, pricing information is presented for purchase, and maintenance as well as warranty terms and conditions.
3.3 Database operations and formats

After the text of the articles is scanned it is stored in a database management system. CDS/ISIS for Windows (WINISIS) serves as the backend database to store the data. The choice of the bibliographic data elements and their tags are according to the Common Communication Format (CCF), to the extent possible. However, it is to be decided whether the Dublin Core (DC) elements set should be adhered to. Of course, this is subject to the point that these elements are deemed as the necessary and sufficient set.

The field definition table for the data elements is given below:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Field</th>
<th>Type</th>
<th>Repeatability</th>
<th>Pattern/Subfield</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title</td>
<td>Alphanumeric</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Creator</td>
<td>Alphanumeric</td>
<td>R</td>
<td>ab</td>
</tr>
<tr>
<td>3</td>
<td>Subject</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>4</td>
<td>Description</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>5</td>
<td>Publisher</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>6</td>
<td>Contributor</td>
<td>Alphanumeric</td>
<td>R</td>
<td>ab</td>
</tr>
<tr>
<td>7</td>
<td>Date</td>
<td>Alphanumeric</td>
<td>R</td>
<td>abmy</td>
</tr>
<tr>
<td>8</td>
<td>Type</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>9</td>
<td>Format</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>10</td>
<td>Identifier</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>11</td>
<td>Source</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>12</td>
<td>Language</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>13</td>
<td>Relation</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>14</td>
<td>Coverage</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
<tr>
<td>15</td>
<td>Rights</td>
<td>Alphanumeric</td>
<td>R</td>
<td>a</td>
</tr>
</tbody>
</table>

Fig. 3: WINISIS Field Definition Table

For every article, the bibliographic entries with data elements as shown are made. The actual full text of the document is stored in the PDF format and a link is created to it. The PDF files are generated using the Adobe Acrobat writer package.

Why PDF?

On the Internet the most popular format is HTML files, sometimes even referred as HTML databases though there are only links from one document to another. The other ways to disseminate information, especially articles, are by 1. postscript files and 2. PDF format. The postscript files, are files that contain the coding that printers understand for printing the document properly. These files can be opened and edited but are to be printed using GhostView and GhostScript software. The reasons for choosing the PDF format here are the following:

PDF preserves the format of the document as in print form.

1. This satisfies the familiar complaint of users that “e-documents are not quite document like”.

14
2. It enables the proper referencing of the articles. That is, the authors can still mention the volume, page number, etc. of the document even while referring the electronic form of the article/document.

3. Acrobat Reader is more popular than GhostView and PDF files are becoming more and more popular than PostScript files.

### 3.4 Design of the retrieval mechanism

For efficient retrieval, the database should have efficient indexing. In some of the seminars the authors of the various articles have supplied the keywords describing the articles. In such cases, if the author-provided keywords are found adequate for retrieval purposes, they may be taken as descriptors. In any case, for the other articles, descriptors can be assigned manually by content analysis. However, CDS/ISIS has many good indexing techniques to automatically generate access points.

### 3.5 User interface --web interface/ browsing and search facility

The system is only good when with the least interference of any party, the end user retrieves the data required at the time of need. For this, the system should have an efficient user interface. A study to point out the most desirable features to be included in the user interface to a Digital Library may be undertaken. However the very popular user interfaces today in the Internet environment are the web browsers. Whether the document is online on web or off-line, hypertext and hypermedia is the common way of presenting information and it probably is here to stay.

Hence, for the DRTC Digital Library the user interface is provided through the web browsers. The web interface is achieved by the WWWISIS package. The documentation of WWWISIS has given directions to write the required CGI (Common Gateway Interface) scripts for creating the user interface.

The CGI works as an intermediary between the browser and the web server. The requests from the browser to the server are sent through CGI scripts. Examples of CGI scripting languages are Perl, Tcl, JAVA Scripts etc. The CGI scripts forward the requests from the browser to the web server, and also retrieve data from the server and send them to the browser in HTML format. These CGI programs are normally kept in ‘cgi-bin’ directory.

The CGI programs therefore:

1) Collect request from the browser
2) And send data back to the browser in HTML format

The most common method of collecting data from the browser is to use the HTML tag ‘FORM’. The FORM tag contains another element called ‘ACTION’ where we can
Specify the action (i.e. the program) to be invoked. Syntax of the ‘ACTION’ tag is given as

\[
\text{\texttt{<FORM ACTION=http://127.0.0.1/cgi-bin/dlib.bat METHOD="POST">}}
\]

**Steps in setting up wwwisis interface**

1. Create a HTML file, which serves as the first search interface. This may be named ‘index.html’, it should be placed in ‘\frontp~1\content\dlib’. *(Refer appendix II for samples of the files mentioned in this section)*

2. Create a program that is to be invoked from the above file. If we call this program as ‘dlib.bat’, it should be used with ‘ACTION’ of ‘FORM’ tag in the index.html file and should be in the ‘\frontp~1\content\cgi-bin’ directory.

3. Create ‘dlib.cgi’ file to present the various options for the ‘wwwi32’ command used in ‘dlib.bat’ file. The ‘dlib.cgi’ in turn refers to various other files. This file also should be in the ‘\frontp~1\content\cgi-bin’ directory.

4. Copy ‘dlib.fst’ file to ‘\frontp~1\content\cgi-bin’

5. Create ‘head.pft’ file

6. Create ‘dlib.pft’ file

7. Create ‘tail.pft’ file

* *(Refer appendix II for the actual scripts)*

All the above files should be in ‘\frontp~1\content\cgi-bin’, except the ‘index.html’ file, which should be in ‘\frontpage\content\dlib’.

Now open the browser like Netscape Navigator and enter the URL something similar to the following

\[
http://isibang.ac.in/dlib/
\]

If you do not have either an alphabetical or Numeric IP address or even an Ethernet card the following URL should work on any web server, though it can not be accessed from some other systems

\[
http://127.0.0.1/dlib/
\]

This URL leads the user to the search interface. The interface provides both browsing facility and searching by keywords.
Browsing features

This page has enlisted the DRTC seminar volumes by the dates and the year along with links to the content pages of each. The titles of the articles are linked to the actual documents. For convenience, the HTML index with dates is provided at the top of the page so that it is easy to access the relevant year without much scrolling.

Searching options

Search may be conducted using a keyword or a set of keywords. Also the different fields such as Author, Title etc., may be selected in Boolean combination of other fields for searching a particular term. Even the display can be customized to display only required fields. Customization with number of records per screen and the colour of the background can also be done through the on screen options provided.

Description

Users of the Net are at large trying to find out the relevant information with the existing search engines. Now accepting this problem and looking for solutions leads us to the established techniques of document description. That is, if there were a document description for each Internet resource then the search engines could be made powerful whereby they could search with semantics and context of the search terms. Now, who takes the responsibility of describing the documents is another question. In a distributed and varied information system such as the Internet, the responsibility shifts from an intermediary to the creator (2). In any case, accepting the fact someone does the exercise, what results from it is Metadata.

Metadata is structured data about data. The term has been used in recent years, and has become particularly common with the popularity of the World Wide Web. However, the underlying concepts have been in use for a long time, since collections of information have been organized for a long time. Library catalogs represent a well-established variety of metadata that has served for decades as collection management and resource discovery tools (3).

Dublin Core metadata elements represent a broad, interdisciplinary consensus about the core set of elements that are likely to be widely useful to support resource discovery. (The name comes from the original core-element-building workshop held in Dublin, Ohio.) (3). The Dublin Core set of elements are as given in the table below.

<table>
<thead>
<tr>
<th>Content</th>
<th>Intellectual Property</th>
<th>Instantiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Creator (Author)</td>
<td>Date</td>
</tr>
<tr>
<td>Subject</td>
<td>Publisher</td>
<td>Type</td>
</tr>
<tr>
<td>Description</td>
<td>Contributor (Other agent)</td>
<td>Format (Form)</td>
</tr>
<tr>
<td>Source</td>
<td>Rights*</td>
<td>Identifier</td>
</tr>
<tr>
<td>Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coverage</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4: Dublin core elements

* indicates the two new elements which were added on to the thirteen basic set.
The document description of the official website for Dublin core discussions is given below:

Title: A DC Approach to Display Retrieved Web Resources
Creator: Devika P. Madalli
Description: Unorganized distribution of resources on Internet has created an environment of uncertainty for ‘Information searchers’. The existing search strategies and web searching lack in semantics while fetching required information with lot of noise. Added to this is the confusing display of search results, which makes situation even worse. The paper discusses the importance of display formats for search results while taking a critical look at present AACR2 rules for display of Internet resources. Further chalks out ‘search result display’ using Dublin Core elements.
Date: 2001-02-26

3.7 Display

Since the above elements are fairly expressive of the data they contain, a model is suggested below for display of the retrieved web documents in terms of Dublin Core elements. In general, the format is fixed with the elements’ name being printed for all documents against which the actual data contained is presented. This approach has been followed in CD-ROM databases as well as Online databases. Many a time the elements names are in abbreviated form like AU for Author, TI for Title etc. The next feature is for flexibility in display elements. Option is provided for the users to select the elements they want to display. Only those selected will be displayed. Further if a search is conducted by a particular element then in the display that element and data against it are highlighted. Further (optionally this feature may be added) if data against a particular element is not found then the element name is printed with the caption ‘data not available’ against it (Appendix I).

Coding for DC in HTML documents

The HTML encoding allows elements of DC metadata to be interspersed with non-DC elements (provided such mixing is consistent with rules governing use of those non-DC elements). A DC element is indicated by the prefix "DC", and a non-DC element by some other prefix. (4)

The META tag

The META tag of HTML is designed to encode a named metadata element. Each element describes a given aspect of a document or other information resource. For example, this tagged metadata element,

<meta name = "DC.Creator" content = "Simpson, Homer">
says that Homer Simpson is the Creator, where the element named Creator is defined in the DC element set. In the more general form,

\[
\text{<meta name } = "\text{PREFIX.ELEMENT_NAME}"
content = "\text{ELEMENT_VALUE}>"
\]

the capitalized words are meant to be replaced in actual descriptions; thus in the example,

\[
\text{ELEMENT_NAME } \text{ was: Creator}
\text{ELEMENT_VALUE } \text{ was: Simpson, Homer}
\text{and PREFIX } \text{ was: DC}
\]

Within a META tag the first letter of a Dublin Core element name is capitalized. DC places no restriction on alphabetic case in an element value and any number of META tagged elements may appear together, in any order. More than one DC element with the same name may appear, and each DC element is optional.

**Semantic approach using XML**

XML stands for eXtensible Markup Language and is the universal format for structured documents and data on the Web. XML allows the creator of web resources to add tags as required. For example for bibliographic description we need an element called *Author*. In XML it may be used as `<author>` tag and may be written as follows:

\[
<\text{author}> \text{S.R. Ranganathan}</author>
\]

The use of XML promises to enhance the search engines with semantic searching capabilities. XML documents maybe viewed as a collection of text documents with additional tags and relations between these tags. Integrating information retrieval and XML search techniques will enable more sophisticated search on the structure as well as the content of documents while also enabling exercises like document similarity ranking and keyword search. Resource Description Framework (RDF) recommended by the World Wide Web Consortium (W3C) was proposed to model meta-data about the resources of the web. RDF promises to create an information backbone into which many diverse information sources can be connected. With every source representing information in the same way, the prospect is that structured queries over the whole Web become possible (4).

**Example of DC embedded HTML for metadata (5)**

\[
<\text{link}
rel="\text{schema.DC}"
href="http://dublincore.org/qdcmes/1.0/"
title="\text{DCMES plus DCMI recommended qualifiers}>"
<\text{meta name="\text{DC.Element} content="Unqualified value}>"
<\text{meta name="\text{DC.Element.ER} scheme="schemeA content="Value coded according to schemeA}"
<\text{meta name="\text{DC.Element.ER} scheme="listB content="Value selected}"
\]
Where the codewords are:

- Element is one of the 15 DCMES Elements,
- ER represents an Element Refinement,
- schemeA is an encoding scheme,
- listB is a controlled vocabulary,
- langC is a language-code,

Example of XML format for resource description using DC

```xml
<?xml version="1.0" ?>
  <rdf:Description about="http://www.isibang.ac.in/drtc/drtcdl/dev.htm">
    <dc:title>A DC approach to display retrieved web resources</dc:title>
    <dc:creator>Devika P. Madalli</dc:creator>
    <dc:description>Unorganized distribution of resources on Internet has created an environment of uncertainty for ‘Information searchers’. The existing search strategies and web searching lack in semantics while fetching required information with lot of noise. Added to this is the confusing display of search results, which makes situation even worse. The paper discusses the importance of display formats for search results while taking a critical look at present AACR2 rules for display of Internet resources. Further chalks out ‘search result display’ using Dublin Core elements.</dc:description>
    <dc:date>2001-02-26</dc:date>
    <dc:format>text/html</dc:format>
    <dc:language>en</dc:language>
    <dc:publisher>Documentation Research and Training Centre</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

3.8 Incorporate interactivity and contextual help

The main features required for the users to access and carry out a successful search are elaborate directions and help. It is proposed to have online help in addition to technical documentation. To incorporate interactivity and navigational features multimedia components may be used in the online help package. Of course the help should be context sensitive in the sense that from any stage of the search process the help pertaining to that particular stage may be presented.
4 Conclusion

There are various questions that arise when a digital library is hosted. At the forefront, the issues must be addressed with comprehensive and withstanding Policy Formulation. In the design aspect, the desirable technologies to be adapted have been presented in the above sections. However, it is to be decided whether the data will be entered according to the fields as prescribed by the CCF for bibliographic data or by the Dublin Core. Though the Dublin Core is a very limited set of items and may appear easy to adapt to, the set will not suffice for efficient retrieval. Information Retrieval Systems have already treaded this much beaten path. Bibliographic data poses its own challenges to retrieval systems because of the intrinsic peculiarities. In any case, a serious thought has to be given to the issue before the choice of the data elements for a project such as building a digital library is made.

Further, administrative and institutional policies regarding the provision of access to the DL, membership details and dissemination of the document are a few considerations. The basic objective of the DRTC seminars and conferences is to focus on various tools; techniques for information work with active participation from the information professionals who share their experiences. It is also the aim to provide wide access to scholarly publications. The DRTC Digital Library would be an important step to achieving this.

5 References

2. MADALLI (Devika P.). A DC approach to display retrieved web resources. IN Workshop on Multimedia and Internet Technologies, DRTC : Bangalore, 2001, Paper AA.
5. DUMBILL (Edd). Putting RDF to work. {8/09/00}. http://www.xml.com/pub/a/2000/08/09/rdfdb/
Appendix I

Display by DC elements

Query String: **AACR2 and Dublin Core**

**Record 1**

<table>
<thead>
<tr>
<th>Title:</th>
<th>A DC approach to display retrieved web resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creator:</td>
<td>Devika P. Madalli</td>
</tr>
<tr>
<td>Subject:</td>
<td>Metadata</td>
</tr>
<tr>
<td>Description:</td>
<td>This article deals with the Internet search results display according to Dublin Core Elements</td>
</tr>
<tr>
<td>Publisher:</td>
<td>DRTC</td>
</tr>
<tr>
<td>Date:</td>
<td>2001-02-26</td>
</tr>
<tr>
<td>Type:</td>
<td>Article <em>(ideally should be taken from a list prescribed for the purpose)</em></td>
</tr>
<tr>
<td>Format:</td>
<td>Web resource</td>
</tr>
<tr>
<td>Identifier:</td>
<td><a href="http://www.isibang.ac.in/drtc/faculty/devika/publications/aacrdc.html">www.isibang.ac.in/drtc/faculty/devika/publications/aacrdc.html</a></td>
</tr>
<tr>
<td>Source:</td>
<td>DRTC Seminar Proceedings 2001</td>
</tr>
<tr>
<td>Language:</td>
<td>En</td>
</tr>
<tr>
<td>Rights:</td>
<td>Indian Statistical Institute</td>
</tr>
</tbody>
</table>

**Record 2**

<table>
<thead>
<tr>
<th>Title:</th>
<th>Dublin Core and Serials: How Well Do the DC Data Elements Describe E-Serials?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creator:</td>
<td>Wayne Jones</td>
</tr>
<tr>
<td>Subject:</td>
<td>Dublin Core, E-Serials</td>
</tr>
<tr>
<td>Description:</td>
<td>This article deals with Dublin Core elements for electronic serials</td>
</tr>
<tr>
<td>Publisher:</td>
<td>MIT Library</td>
</tr>
<tr>
<td>Date:</td>
<td>1997-11-10</td>
</tr>
<tr>
<td>Type:</td>
<td>Article</td>
</tr>
<tr>
<td>Format:</td>
<td>Web resource</td>
</tr>
<tr>
<td>Source:</td>
<td>Distance Learning Project done as part of the OCLC Institute's 1st Knowledge Access Management seminar</td>
</tr>
<tr>
<td>Language:</td>
<td>En</td>
</tr>
<tr>
<td>Rights:</td>
<td>MIT</td>
</tr>
</tbody>
</table>
Fig. 5 Search Display by DC Elements
Appendix II

Creation of the *index.html*

This step involves in the creation of the first file i.e. ‘index.html’, if this file is in ‘frontp~1\content\library’ directory, one can invoke this file from the browser by enter the following URL

`http://isibang.ac.in/dlib/`

The complete listing of the ‘index.html’ file is given below.

File: index.html

```html
<html>
<head>
<title>TEST</title>
</head>
<body>
<h1>OPAC OF THE LIBRARY</h1>
<form method="get" action="/cgi-bin/dlib.bat/">
<p><input type="text" name="tag2001" value=" " size="20">

<select name="tag2002" size="1">
   <option value=" * ">AND</option>
   <option value=" + ">OR</option>
   <option value="~">NOT</option>
</select>
</p>
<p><input type="text" name="tag2003" value=" " size="20">

<select name="field" size="1">
   <option value="">ALL</option>
   <option value="24">Title</option>
   <option value="70">Author</option>
</select>
</p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
   <input type="submit" name="button" value="submit">
</form>

</body>
</html>
```

The ‘dlib.bat’ file is mentioned as the action to be taken once the ‘submit’ button is pressed. The ‘submit’ button should be clicked on once the search elements are entered. The “input” tag of HTML describes that the input is “text” TYPE, and the “VALUE” is presently blank and once the value is entered, it will be captured in the variable called ‘tag2001’. Similarly, the next lines in the file describe the input for the tag2002, tag2003 and ‘field’. That is, the tag2001 captures the input of the first key
word, tag2002 captures the choice of Boolean operator like ‘and’, ‘or’, ‘not’; tag2003 captures the next keyword and ‘field’ captures the information whether the keywords should be searched in all fields or only in title or author fields.

In other words, the FORM tag of HTML passes the information you have entered once the ‘submit’ button is pressed.

dlib.bat File

The is the batch file which invokes the command wwwi32

Listing of dlib.bat

wwwi32 pfxtag=tag cgi=@dlib.cgi

In the above program various parameters to wwwi32 program are used. The ‘pfxtag’ parameter is meant that the variables ‘tag2001’, ‘tag2002’, ‘tag2003’ should be treated as tags v2001, v2002, v2003 in the virtual record of CDS/ISIS. Here, it should be noted that the most interesting part of wwwisis is that it generates virtual records out of each record generated from CDS/ISIS database. The present paper uses the DRTC database. In addition to fields like Author, Title etc., each record will contain fields 2001, 2002, 2003 with the information/data we have entered. The ‘pfxtag’ indicates that we are using the word ‘tag’ as prefix to the actual tag. It is a good idea to use tags greater than 2000 for the simple reason they will not conflict with the database tags as the data base tags mostly contain 3 digits and the tags greater than 1000 and less than 2000 are used by wwwisis to hold various other kinds of information, like number of records retrieved (tag 1002), the Boolean expression (tag 1021) etc.

The above program ‘dlib.bat’ in turn refers to another CGI file called ‘dlib.cgi’. The contents of ‘dlib.cgi’ are given below:

Listing of file dlib.cgi

'\db=dlib'/
'\bool=',
{
  if p(v2001) then v2001, fi
  if p(v2003) then v2002, v2003,fi
}/,
'\prolog=@head.pft'/
'\pft=@dlib.pft'/
'\epilog=@tail.pft'/

In the above program:
The ‘db=dlib’ states that the database to be used is ‘dlib’.
The ‘bool=’ states that the Boolean search expression is to be prepared from the contents of tag v2001, v2002, v2003.

The ‘prolog=’ states that the output HTML file should contain the contents of ‘head.pft’ as the first few statements of HTML file.

The ‘pft=’ states that ‘dlib.pft’ file is to used for the actual display of the records

The ‘epilog=’ states that the ‘tail.pft’ should be appended to ‘dlib.pft’ file

File head.pft

The following lines will be prefixing the dlib.pft

```html
<html> /
<title>test</title> /
<body> /
<h1>Search Results</h1>
```

Notes on dlib.pft

This is the main display format with the necessary HTML tags. In the following file,

```html
if val(v1001) = 1 then 'Total Found: ', v1002,fi/
```

displays the number of records retrieved for the present query by displaying the tag v1002.

File dlib.pft

```html
mhl,'<TABLE WIDTH="100%" BORDER=0>'
mhl,'<TR><TD WIDTH="100%">Record N. ',mfn(1),'</TD></TR></TABLE>'
mhl,'<TABLE WIDTH="100%" BORDER>'
if p(v1) then '<TR><TD WIDTH="30%">Title </TD><TD>',v1,'</TD></TR>'/fi,
if p(v2) then '<TR><TD WIDTH="30%">Creator </TD><TD>',v2+”; |’</TD></TR>',fi,/ 
if p(v3) then '<TR><TD WIDTH="30%">Subject </TD><TD>',v3+”; |’</TD></TR>',fi,/ 
if p(v4) then '<TR><TD WIDTH="30%">Description </TD><TD>',v4+|; |’</TD></TR>',fi,/ 
if p(v5) then '<TR><TD WIDTH="30%">Publisher </TD><TD>',v5+|; |’</TD></TR>',fi,/ 
if p(v6) then '<TR><TD WIDTH="30%">Contributor </TD><TD>',v6+|; |’</TD></TR>',fi,/ 
if p(v7) then '<TR><TD WIDTH="30%">Date </TD><TD>',v7+|; |’</TD></TR>',fi,/ 
if p(v8) then '<TR><TD WIDTH="30%">Type </TD><TD>',v8+|; |’</TD></TR>',fi,/ 
```
if p(v9) then '<TR><TD WIDTH="30%"><I>Format</I></TD><TD>',v9+; |'</TD></TR>',fi,/mpl,
if p(v10) then '<TR><TD WIDTH="30%"><I>Identifier (Full text)</I></TD><TD>',('<A HREF="/dlib/pdfs/''v7^y'', v10^a, ''http://ard.isibang.ac.in/dlib/pdfs/'',v7^y,''/'',v10^a''/''A''), '</TD></TR>' ,fi,/mhl,
if p(v11) then '<TR><TD WIDTH="30%"><I>Source</I></TD><TD>',v11+; |'</TD></TR>',fi,/if p(v12) then '<TR><TD WIDTH="30%"><I>Language</I></TD><TD>',v12+; |'</TD></TR>',fi,/if p(v13) then '<TR><TD WIDTH="30%"><I>Relation</I></TD><TD>',v13+; |'</TD></TR>',fi,/if p(v14) then '<TR><TD WIDTH="30%"><I>Coverage</I></TD><TD>',v14+; |'</TD></TR>',fi,/if p(v15) then '<TR><TD WIDTH="30%"><I>Rights</I></TD><TD>',v15+; |'</TD></TR>',fi,/'</TABLE><P>

File: tail.pft

if (v1091 = '7') then 'No ----Records retrieved'fi,/
if (v1091 = '0') then 'END OF SEARCH RESULTS', fi,/

27