Design and development of a user interface for the library's database at ICRISAT

P. RATNAKUMAR and L.J. HARAVU

ABSTRACT

This paper describes the development of a user interface to a composite database, i.e. one containing records for library-held monographs as well as for journal articles, conference documents, and non-conventional literature, downloaded from global databases. The user environment and characteristics of users for whom the interface is targeted are discussed. The features of BASIS, the retrieval software in use at ICRISAT to create, maintain and search the library's database and the need to build an interface for the occasional user are discussed. The objectives of building the user interface, software options considered, and the development of the interface using the BASIS menu language are presented.

1. Introduction

In an automated library environment, users' information needs are met by the library's online catalogue or information retrieval system. However, the user needs to use the library's online catalogue or database relatively occasionally and may be unwilling to learn and remember the command language and syntax of the information retrieval system or online public access catalogue (OPAC). User-friendly front ends to library databases and online catalogues have been the subject of much effort and research in recent years[1,2,3,4,5,6].

A user interface may be defined as an interactive system through which the lay user can communicate with the software irrespective of its complexity, with little or no knowledge of it, and easily exploit/utilize its capabilities. User interfaces available in today's market can be categorized as:

- menu-driven systems;
- command-based systems;
- direct manipulation systems;
- systems based on artificial intelligence using expert systems methodologies[7].

In menu-driven systems, text menus provide the user with a list of options and selection may be made either by typing a number or character representing the option, or by simply pressing the Enter key when a particular option is highlighted. Graphical menus are based on symbols, icons, colours, or patterns to represent an action/process instead of text or descriptions of options. Selection is made by a graphic device such as a mouse or graphic tablet. Graphic/text menu systems select an operation first followed by the object on which the action is to be carried out.

Program, vol. 28, no. 1, January 1994, pp. 15–27

Command-driven systems depend on the user's memory. These systems provide the user with a prompt to which a response is made either by a command line with necessary parameters or a file name with user-defined commands. UNIX is a good example which allows the user to activate a command at the prompt or call one of the user's shell programs. Shell programming techniques allow the user to develop menu-driven systems also. Database management system (DBMS) packages like dBASE IV provide the user with the facility of developing a multi-layered menu-driven system and also to use commands directly at the system prompt.

Direct manipulation systems are object oriented in the sense that an object is chosen first, upon which an operation is performed. These systems are based on the assumption that "it is easier to see something and point to it than to remember and type it"[8]. A cursor is used to indicate the point selected on the screen, and different shapes of cursors are used to indicate the current mode, function, or action being performed. Direct manipulation interfaces are helpful in minimising the user's effort in learning commands, in eliminating typographic errors and in keeping the attention of the user on screen displays. Examples of direct manipulation systems include X- Windows and the Macintosh HyperCard.

Well known information storage and retrieval (ISR) systems (Dialog, BRS, ORBIT) all began as command-driven systems. These were used primarily by information specialists rather than by the end user. However, with the emergence of minicomputer and microcomputer-based systems and the possibility of distributed access and processing in wide and local area networks, the need for end user access to information retrieval systems has come to the fore in the last decade.

Many integrated library management systems (Geac, VTLS, CLSI)[9] provide an OPAC module. OPACs deal primarily with monographic material that is traditionally held by libraries and described in card catalogues. Many OPACs use menu-driven interfaces together with help messages. Subject searching in OPACs proceeds from the display of pre-coordinated subject headings that were assigned to bibliographic items in the online catalogue. The user is then allowed to choose one or more headings, which are then used to retrieve the required bibliographic record(s). Some OPACs permit boolean combinations of selected headings. OPACs differ from ISR systems in that subject searching in ISR systems is based on post-coordinate index principles. ISR systems, generally speaking, allow greater flexibility and freedom than OPACs to combine terms representing subjects . However, OPACs provide access not only to a library's holdings but also let the user know if the publication is out on loan, or if it is on order.

Retrieval software which uses relational database management systems have used Structured Query Language (SQL) as the query language[10,11]. Form-based interfaces that enable users to specify their search parameter(s) have been used with such systems. The filled-in form becomes the input to routines that generate a SQL query which is then executed transparently to the user.

Concepts of client-server systems have had their impact on the design of OPACs[12]. Some OPACs in the UK and USA have begun to use graphical user interfaces to OPACs as part of the so called Campus Wide Information Systems (CWIS) that are beginning to emerge in university campuses.

This paper presents our attempt to develop a user interface to a composite database, i.e. one which contains records for library-held monographs as well as for journal articles, conference documents, and non-conventional literature, downloaded from global databases. The database and user environment in which the interface will be used are described as well as the philosophy used in designing the interface.

2. ICRISAT library's database

The International Crops Research Institute for the Semi-Arid Tropics (ICRISAT) library's database is a specialised one and uses data drawn from two international databases: AGRIS (International Information System for Agricultural Sciences and Technology) of the Food and Agricultural Organization (FAO) and CAB International (CABI). CABI provides a subset of their database each month based on a macro-profile of crop, subject and other interests that were communicated to them. The AGRIS tape received each month is scanned by a program developed inhouse to identify and download records found relevant to interests at ICRISAT. Since data are drawn from two sources, the descriptors from these two databases are stored in separate fields, both of which are searchable.

Bibliographic data for books, book chapters, conference documents and non-conventional literature added to the library, forms the third source of input to the database. The database contains 35,000 bibliographic records on ICRISAT's mandate crops (sorghum, pearl millet, pigeonpea, chickpea and groundnut) and related areas such as agroclimatology, biotechnology, farming systems, soils of the semi-arid tropics (SAT), the agricultural economy of SAT countries, etc. It is meant to serve both as an online library catalogue and as an ISR system for specialised subject retrieval. The database resides on ICRISAT's MicroVax 3900. The software used to create, maintain and search the database is BASIS (Battelle Automated Search Information System).

3. BASIS and its query language

BASIS is a traditional ISR software package. Databases under BASIS are stored in a single logical file with several secondary key indexes for quick retrieval. A number of commands allow the user to query the database. The most frequently used command to retrieve information from the database is FIND. The user needs to use a prefix or suffix, pre-defined in the Data Definition Language of BASIS, to specify the field in which the search is required to be done. BASIS supports range searching of numeric fields, relational operators, term truncation, and, of course, boolean expressions. It is also possible to search fields which are not indexed. Like other well known ISR software, BASIS also provides the capability to make selections of search terms after browsing through one or more inverted files created for a given database.

Each search expression creates a set of retrieved records which is uniquely identified by a set number. A user can create upto 2,000 sets during a search session.

These sets can be LISTed at any time during a search session and can be combined using boolean operators. Sets created in a session can be saved to disc and restored, if required in future sessions. Like in other command-driven software, all the commands of BASIS have a specific syntax and associated parameters which are required to be followed strictly. The help facility in BASIS is not particularly useful to an occasional user.

BASIS permits the writing of programs in high-level languages (HLL) like Fortran, Pascal, etc., which can call BASIS modules and access database records. HLL programs may also be written as user exits to be called from within BASIS modules. One of BASIS' modules is the Profile (Procedures File) module which enables the writing of a series of search or other statements as a file that can be called and executed from within BASIS or from a HLL program. BASIS also provides a programming language called Basis Menu Language (BML) that can be used to create a Profile file, typically to create a menu-driven interface to one or more BASIS databases. BASIS also provides a Report generator language which enables the writing of programs to produce formatted outputs. It became quite clear from the outset that it would be unrealistic to expect end users (scientists, economists, research assistants, administrators) to query the database on their own using the command language of BASIS. Our interactions with users at ICRISAT revealed that they were keen to do their own searches but understandably were reluctant to learn the command language of BASIS.

4. Users at ICRISAT

The interface described in this paper is targeted primarily at agricultural researchers drawn from both the developed and developing world. Like in all specialist environments, users at ICRISAT need to be able to browse generally through the database, as well as to query it to retrieve information on specific topics. Access to the database is required predominantly for research planning, literature reviews, problem solving, corroboration of research results and for new ideas. In addition, there is also the need to satisfy specific document location and delivery needs. It became clear that information retrieval needs rather than library catalogue consultation needs should drive the design of the interface.

In informal interactions with users it became clear that they would be interested in a system that would enable them to search titles and abstracts of documents for specific words or phrases of their choice, apart from being able to use thesaural descriptors. Users were of the view that thesaural descriptors alone often did not meet their needs adequately enough. The reason for this (which we believe is equally true for other similar environments) is the difficulty of articulating needs *a priori* in precise terms. Another factor is that the user is looking for information on abstract ideas that cannot be precisely specified.

It was also found that there was confusion in the minds of users about the boolean operators OR and AND. Many users relate these operators to arithmetical addition and multiplication respectively. We found that users sometimes used the AND operator when in fact they needed to use the OR operator. Since the proper use of boolean expressions is essential to relevant retrieval in BASIS, it became clear that the interface must enable the user to make boolean combinations intuitively.

Users at ICRISAT, by and large, are considerably exposed to computer-based systems. Many of them work both with PC-based as well as mainframe-based software. Statistical analysis, word processing, maintenance of personal bibliographic databases, access to the Institute's germplasm research projects, and mailing list databases are some of the computer-based systems that users are quite comfortable with. Lotus 1-2-3, dBase III+, Foxpro2, Genstat, and Wordperfect software are quite widely used within the Institute. The proliferation of PCs within the Institute and the establishment of a local area network (LAN) linking PCs to the Institute's MicroVax cluster has not only improved computing capacity but has also raised expectations of what service departments such as the library should do to enable their products and services to become available more widely and easily. Like in the developed world, the friendliness of PC-based systems and the increasing familiarity of users with these, inevitably has conditioned expectations about interfaces to applications.

5. Software options

The following options to build the interface to the library's database were considered:

- use a HLL like Fortran or Pascal to write the routines for the interface;
- use Microsoft Windows together with remote procedure calls to build an interface that could be run from microcomputers on the Institute's LAN. Such an approach is similar to the client-server approach in that client processes communicate with a database server, using graphical user interfaces (GUIs);
- write the interface using the menu language provided by BASIS, viz., BML.

The first and third options could be run both from dumb terminals as well as from microcomputers on the LAN, while the second option could only be run from microcomputers on the LAN.

A limited experiment was done in writing Fortran programs to build an interface. It was found that the time and effort involved would not necessarily be commensurate with the quality of the end-product. The second alternative appeared to be the most attractive in terms of the quality and friendliness of the product. However, skills available in the ICRISAT library were not adequate to try this approach. Computer centre staff who were approached for help were over-committed to other tasks. However, their help in the future has been promised and the development of a Windows-based interface is earmarked for 1994-1995. In the meanwhile, it was decided to work on the development of the interface using the third alternative, viz. to write routines in BML.

The objectives of the interface are to:

- encourage end users to do their own searches;
- eliminate the need to maintain the card catalogue and utilise the professional manpower so relieved for other tasks;
- reduce the work load on professionals in meeting the growing demand for searches.

6. Interface characteristics

6.1 User-based criteria

Based on the discussions with users, the following criteria were considered to be important in designing the interface:

- The interface should be simple to use and understand.
- It should provide help messages, wherever appropriate.
- It should allow the use of the three boolean operators (AND, OR, AND NOT) during a search in as transparent a manner as possible.
- The user should be allowed to select relevant records from a search set while browsing through the items in that set, i.e. the users should be able to select and mark records that they want as final output or as an intermediate search set.
- The user should be allowed to traverse through the records of a search set in any order one may choose to do so, i.e., sequentially, at random, or by record number.
- The user should be able to perform the three traditional ways of searching a catalogue (by author, title and subject); in addition, users should have the option to do a simple two parameter search (adequate in author and author-title searches), or a detailed search which would involve several sets of terms in OR, AND, and AND NOT combinations.
- It should allow the user to delimit the search by one or more types of documents (journal articles, monographs, conference papers, theses etc.) and/or by the year of publication. Further, the user should be able to restrict the output to library-held materials.
- The user should be able to browse through indexes and select relevant records based on terms in the indexes.
- The user should be able to get a hard copy of the search output in a specified format.
- The user should be able to save search sets to disc so that these could be recalled in a future search session. This is useful when the user wishes to discontinue a search session and return to it without loss of the search results at the time it was discontinued.
- It should allow the expert user to enter BASIS and execute query language commands directly.

6.2 Programmer-based criteria

The interface was written as a number of BML routines. These are stored as BASIS profiles (Procedure files) and called/executed from within BASIS. However, BML has a number of limitations:

- a program in BML cannot exceed 1000 lines;
- a BML statement cannot exceed 160 characters;
- only 10 global and 15 local variables can be used in a BML program;
- a BML variable cannot hold more than 160 characters;
- the only screen/cursor control function provided is to erase screen. This limits the kind of screens that can be used with BML. However, the use of escape sequences are allowed to control screen display and cursor positioning;
- although it is possible for a profile to call and execute another profile, this slows down execution and hence increases response time.

In view of the limitations of BML, it was necessary to adhere to the following criteria.

- the interface should be limited to a minimum number of profiles to enable fast response to queries;
- as far as possible, the program should use routines to avoid the coding of operations that are needed repeatedly. This would enable the profile to use less lines of code and enable fast response.

7. Organisation of the interface

The routines of the interface have been broken up into four profiles.

7.1 The Sign-on message profile

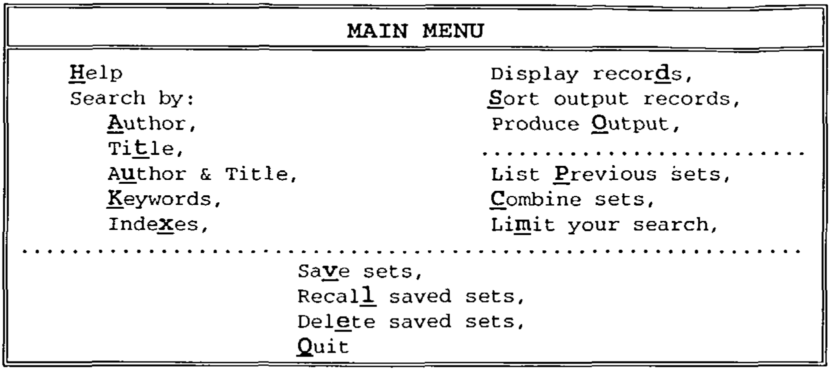
This profile defines the global variables for the entire interface and includes the screen display control variables (reverse video, normal, blink, underline, and cursor position variables), and a message on the scope of the database.

7.2 Help messages profile

Help messages are provided with appropriate examples on the usage of the different search options. Help messages are accessible from the main menu of the search profile.

7.3 Search profile

This is the core profile through which the end user performs the searches. The user enters this profile immediately after the Sign-on message profile. The main menu of the profile is given in Figure 1.



```
                                MAIN MENU

    Help                                   Display records,
    Search by:                             Sort output records,
       Author,                             Produce Output,
       Title,                             .........................
       Author & Title,                     List Previous sets,
       Keywords,                           Combine sets,
       Indexes,                            Limit your search,
.............................................................
                           Save sets,
                           Recall saved sets,
                           Delete saved sets,
                           Quit
```

Figure 1. Main menu

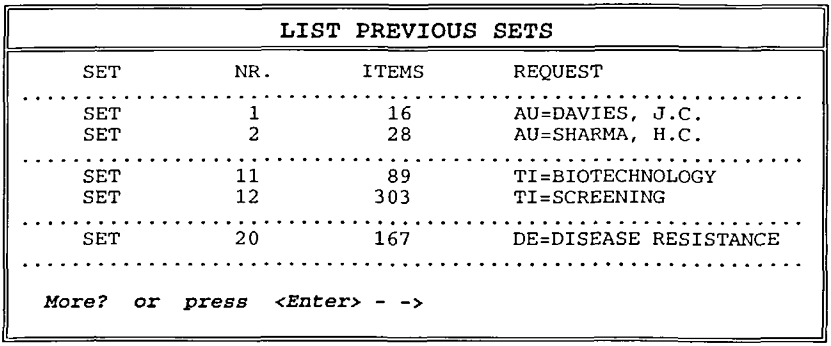
7.3.1 Approaches to implementing boolean searching

Two approaches to implementing boolean searching transparently to the user have been attempted in the interface. In the first approach, the user is provided with a fixed template wherein one is asked to think of the search as comprising two or more search parameters, each of which contains a homogeneous set of terms (i.e., a set of synonymous, related, narrower terms). The OR operator is automatically applied to the terms constituting a search parameter and the AND or AND NOT operator gets applied between the search parameters. In this approach, the resulting search expression generated quite transparently to the user could be as follows:

$$\begin{array}{l}(T_1 \text{ OR } T_{11} \text{ OR } T_{12} \text{ OR} \ldots T_{19}) \text{ AND}\\ (T_2 \text{ OR } T_{21} \text{ OR } T_{22} \text{ OR} \ldots T_{29}) \text{ AND}\\ (T_3 \text{ OR } T_{31} \text{ OR } T_{32} \text{ OR} \ldots T_{39}) \text{ AND/AND NOT}\\ (T_4 \text{ OR } T_{41} \text{ OR } T_{42} \text{ OR} \ldots T_{49})\end{array}$$

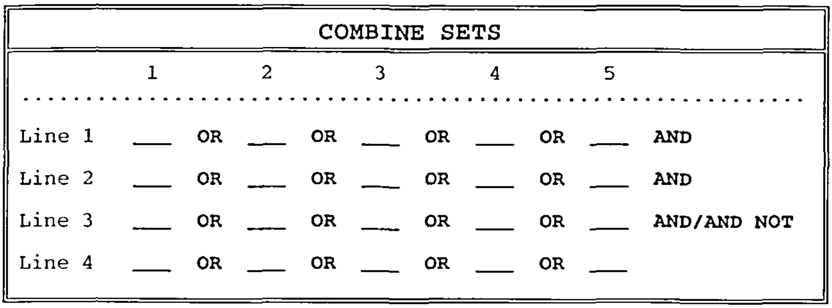
where each parameter enclosed within parentheses comprises a term and up to nine other terms which are equivalent, synonymous, narrower or related terms.

In the second approach, the user is provided with the facility to combine authors/title words/subject terms quite flexibly and in any order. This is possible in the 'Combine Sets' option provided in the main menu given above. This option first lists all the sets created up to that point (as shown in Figure 2) in the search session, and then provides the user with a form which allows one to enter up to 20 set numbers in four lines, in each of which one can enter up to five set numbers (as shown in Figure 3). Here again, the set numbers in each line are ORed into a search parameter and each such parameter is ANDed with the other parameters. In this approach, the user creates sets first and only later thinks of parameterising these and combining them as required.



```
                         LIST PREVIOUS SETS

     SET          NR.         ITEMS         REQUEST
.............................................................
     SET           1            16          AU=DAVIES, J.C.
     SET           2            28          AU=SHARMA, H.C.
.............................................................
     SET          11            89          TI=BIOTECHNOLOGY
     SET          12           303          TI=SCREENING
.............................................................
     SET          20           167          DE=DISEASE RESISTANCE
.............................................................

  More?  or  press  <Enter> - ->
```

Figure 2. Listing previous sets



```
                            COMBINE SETS

               1        2        3        4        5
.............................................................
Line 1        ___  OR  ___  OR  ___  OR  ___  OR  ___  AND

Line 2        ___  OR  ___  OR  ___  OR  ___  OR  ___  AND

Line 3        ___  OR  ___  OR  ___  OR  ___  OR  ___  AND/AND NOT

Line 4        ___  OR  ___  OR  ___  OR  ___  OR  ___
```

Figure 3. Combining sets

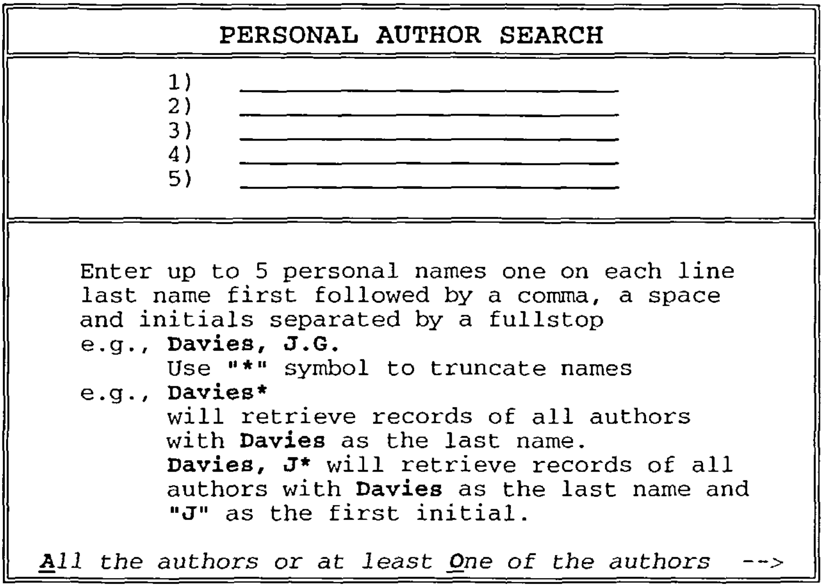
7.3.2 Sub-routines

Operations common to different searches have been identified and are used as sub-routines throughout the profile. The core routines common to all types of operations in the main menu are given below.

a) GET-TERM routine

This routine is invoked in author, title, and subject searches. The user is provided with a screen form with one line for each author or title word. Depending on the kind of search, the routine is called one or more times. In the author, title or simple subject search, the routine is called only once and allows the user to enter up to five author names, title words or subject terms. In such searches, the user can combine names or terms entered either with the OR or AND operators. This is done by asking if the user wants to search for records with any one or more of the names/terms entered, or if the user wants records with all the terms entered.

Depending on the users' response, the OR operator or AND operator is applied as shown in Figure 4.



```
                  PERSONAL AUTHOR SEARCH

                1)  ______________________
                2)  ______________________
                3)  ______________________
                4)  ______________________
                5)  ______________________

     Enter up to 5 personal names one on each line
     last name first followed by a comma, a space
     and initials separated by a fullstop
     e.g., Davies, J.G.
           Use "*" symbol to truncate names
     e.g., Davies*
           will retrieve records of all authors
           with Davies as the last name.
           Davies, J* will retrieve records of all
           authors with Davies as the last name and
           "J" as the first initial.

 All the authors or at least One of the authors  -->
```

Figure 4. Personal author search

In the detailed subject search option, the routine is called up to four times. As described under 7.3.1.

b) DISPLAY-and-SELECT records routine

This routine is invoked immediately after the execution of a search, and displays brief details of records retrieved. The user can browse through the entire set of records in sequence, or at random. Further, the user has the option to see full records. Each record displayed can be marked, i.e. selected as being relevant for output or as an intermediate set. All marked records are stored as a separate set. This routine can also be invoked from the main menu and in such a case it provides the capability to see records retrieved in any previously created set.

c) SAVE-and-RECALL search sets routine

This routine permits the user to save one or more search sets to disc or recall previously saved search sets. The user can save a range of sets, some or all sets one-by-one, or one or more specific sets. The user is asked to assign a mnemonic name for the set or sets that are to be saved.

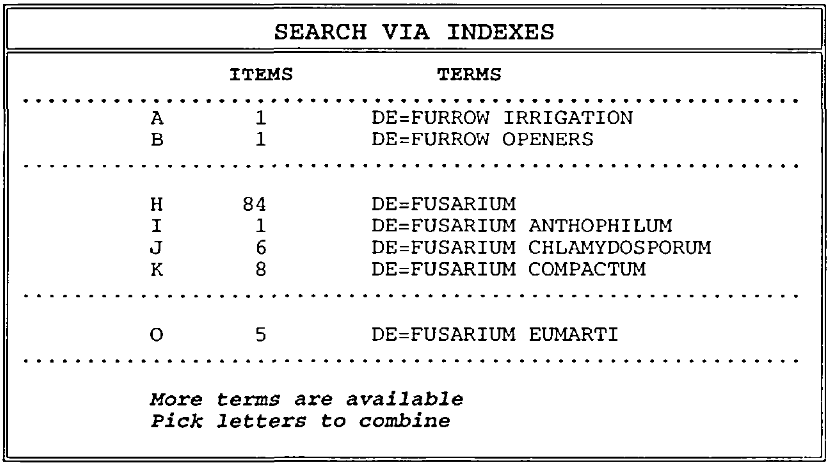
When users want to recall previously saved sets, they are first presented with a list of saved sets by name. The set or sets are recalled by entering the name of the set. Once this is done, the set or sets are available for further manipulation in the continued search session. Similarly, saved sets can be deleted using the delete sets option in the main menu. Typically these options are required when the user wishes to discontinue the search session without loosing the search sets needed when the search is resumed.

7.3.3 Types of searches

In addition to searches by authors, authors and title words, and subjects described above, the interface allows search via indexes directly as a main menu option.

This option provides a selection of four indexes via which the user can formulate the search. These are: Author, Title, Subject and Abstract. The user first selects the index and enters a search word or word stem. A set of 16 matching or near-match terms are displayed from the chosen index along with the number of postings. Each name or term is identified by an alphabetic tag followed by number of postings, prefix of the term, and terms matching input word or stem. For instance, when the user opts to search via the subject index and provides a stem, he is shown terms with the prefix DE, each term being tagged by an alphabetic code as shown as in Figure 5. The user can select one or more terms listed, by referring to such terms using the alphabetic tag and/or proceed with another word or word stem from the same index file. Alternatively, the user can return to the menu to select another index file or return to main menu. The term or terms selected are searched for and stored in one or more sets and are available for further manipulation.

Searching via indexes enables the user to do a form of free-text searching. This method has been found useful in searches which cannot be defined precisely and which require an interactive and iterative search mode to enable the user to formulate the search, and be satisfied that the best result possible has been achieved.

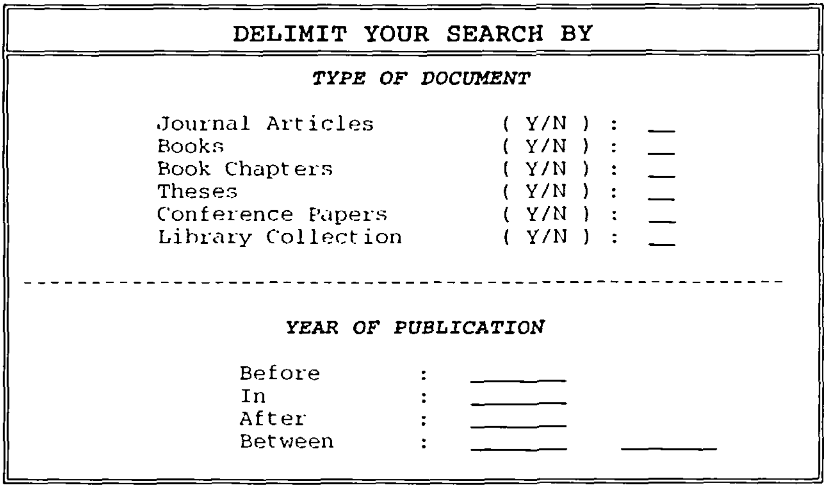


```
                        SEARCH VIA INDEXES

              ITEMS              TERMS
.....................................................
         A        1        DE=FURROW IRRIGATION
         B        1        DE=FURROW OPENERS
.....................................................

         H       84        DE=FUSARIUM
         I        1        DE=FUSARIUM ANTHOPHILUM
         J        6        DE=FUSARIUM CHLAMYDOSPORUM
         K        8        DE=FUSARIUM COMPACTUM
.....................................................

         O        5        DE=FUSARIUM EUMARTI
.....................................................

         More terms are available
         Pick letters to combine
```

Figure 5. Index searching

7.4 DELIMIT-SEARCH profile

This profile allows the user to delimit the search output by type of document (journal articles, monographs, monograph chapters, conference papers, or theses). A screen form is displayed in which the user indicates document types to restrict the output as shown in Figure 6. The user can also restrict the output to documents held by the library. Further, the user can confine the search to documents published before, in, or after a specific year or between two years, e.g. 1987-1992.



```
                     DELIMIT YOUR SEARCH BY

                        TYPE OF DOCUMENT

          Journal Articles            ( Y/N ) :  __
          Books                       ( Y/N ) :  __
          Book Chapters               ( Y/N ) :  __
          Theses                      ( Y/N ) :  __
          Conference Papers           ( Y/N ) :  __
          Library Collection          ( Y/N ) :  __

 ----------------------------------------------------------

                      YEAR OF PUBLICATION

                Before      :  ______
                In          :  ______
                After       :  ______
                Between     :  ______     ______
```

Figure 6. Restricting a search

8. Conclusions

No software is an end in itself. In an evolving area such as user interfaces, there is need to continually evaluate products such as the one described in this paper. This process has just begun at ICRISAT and we believe it will enable us to refine and enhance the application using new concepts and technologies and better understanding of human processes involved in such work.

References

1. D. Harman and L.F. Lunin. Introduction and overview [of perspectives on...human-computer interface]. *Journal of the American Society for Information Science*, vol.43, no.2, March 1992, pp.153-155.

2. C.R. Hildreth. *The on-line catalogue: development and directions.* London: The Library Association. 1989.
3. G. Marchionini. Interfaces for end-user information seeking. *Journal of the American Society for Information Science*, vol.43, no.2, March 1992, pp.156-163.
4. M.M. Yee. System design and cataloging meet the user: user interfaces to online public access catalogs. *Journal of the American Society for Information Science*, vol.42, no.2, March 1991, pp.78-98.
5. M. Hancock-Beaulieu. User friendliness and human-computer interaction in online library catalogues. *Program*, vol.26, no.1, January 1992, pp.29-37.
6. B. Shneiderman. *Designing the user interface: strategies for effective human-computer interaction.* Reading, Massachusetts: Addison-Wesley Publishing Co., 1987, ISBN 0-201-16505-8.
7. S. Gauch. Intelligent information retrieval: an introduction. *Journal of the American Society for Information Science*, vol.43, no.2, March 1992, pp.175-182.
8. J.R. Brown and S. Cunningham. *Programming the user interface: principles and examples.* New York: John Wiley & Sons, Inc., 1989, ISBN 0-471-63843-9.
9. J.R. Matthews. *Public access to online catalogs.* 2nd edition. (Library Automation Planning Guides Series, No.1). New York: Neal-Schuman Publishers, Inc., 1985, ISBN 0-918212-89-8.
10. J.H. Ashford. Text storage and retrieval in the Oracle relational database management system: design study and intended applications. *Program*, vol.21, no.2, April 1987, pp. 108-123.
11. G. Verity. Relational database management systems and open systems used in the development of Oracle libraries. *Program*, vol.27, no.1, January 1993, pp.73-82.
12. G.L. Phillips. Z39.50 and the scholar's workstation concept. *Information Technology and Libraries*, vol.11, no.3, September 1992, pp.261-270.

Authors

P. Ratnakumar, Assistant Library Officer; L.J. Haravu, Sr. Manager, Library & Documentation Services, ICRISAT, Patancheru, A.P. 502324, India.