PRL

TECHNICAL NOTE

TN-85-40

# LIBDBS : A LIBRARY DATA BASE SYSTEM FOR INTERACTIVE STORAGE AND RETRIEVAL OF BIBLIOGRAPHIC INFORMATION.

By
D.P.AHALPARA

February,1985

COMPUTING CENTRE
PHYSICAL RESEARCH LABORATORY
AHMEDABAD-9

## DOCUMENT CONTROL AND DATA SHEET

1. REPORT NO. : PRL : TN : 85-40

2. TITLE AND SUBTITLE

**LIBDBS : A LIBRARY DATA BASE SYSTEM FOR INTERACTIVE STORAGE AND RETRIEVAL OF BIBLIOGRAPHIC INFORMATION**

3. REPORT DATE: FEBRUARY 1,1985

4. TYPE OF REPORT : TECHNICAL REPORT

5. PAGES : 43

6. PRICE : UNPRICED

7. NO. OF REFERENCES: 12

8. AUTHOR(S) : D.P.AHALPARA

9. a) PURPOSE : TO ALLOW FOR STRUCTURED STORAGE AND INTERACTIVE SELECTIVE RETRIEVAL OF BIBLIOGRAPHIC INFORMATION BASED ON USER QUERIES

b) USEFUL FOR: USERS AND STAFF MEMBERS OF A TECHNICAL LIBRARY

10. ORIGINATING UNITS/DIVISIONS AND ADDRESS: COMPUTER CENTRE
PHYSICAL RESEARCH LABORATORY, NAVRANGPURA, AHMEDABAD 380009

11. SPONSORING AGENCY : NIL

12. ABSTRACT : A LIBRARY DATA BASE SYSTEM HAS BEEN DEVELOPED IN PASCAL LANGUAGE. IN A MENUDRIVEN ENVIRONMENT IT ALLOWS FOR SELECTIVE RETRIEVAL OF BIBLIOGRAPHIC INFORMATION IN RESPONSE TO USER QUERIES BASED ON CERTAIN KEYWORDS.

13. KEYWORDS : INFORMATION SYSTEM, LIBRARY AUTOMATION, DBMS

14. DISTRIBUTION STATEMENT : DISTRIBUTION UNLIMITED

15. SECURITY CLASSIFICATION : UNCLASSIFIED

# LIBDBS

## A Library Data Base System for Interactive Storage and Retrieval of Bibliographic Information

D.P.Ahalpara
Computer Centre
Physical Research Laboratory
Navrangpura
Ahmedabad-380009

# INDEX

Abstract

A LIBrary Data Base System (LIBDBS) has been developed for the usage of a Library, allowing for an interactive storage and selective retrieval of bibliographic information based on user queries. The system is made up of about 3,000 lines in Simula language and makes use of Simula's data base management system (DBMS) packages SIMDBM and DBMSET. The core requireme for the storage of the system i s 50K words on the DEC-10 syst

The salient features of the system are as follows:

1) For storage of bibliographic information : Simple a efficient methods are employed allowing for storage in either an interactive mode (useful for small amount of information) or in a batch mode (for a mass storage of huge amount of information).

2) For selective retrieval of information :

a) The Query Processor allows for any easy to learn query language supporting sentences of English language.

b) The retrieval of information may be made based on a variety of keywords.

c) The retrieved information may be selectively accessed in the form of visual display on a terminal, print out on a line printer or storage on a disk file.

# CHAPTER - 1

## INTRODUCTION

a) What is an information system ?

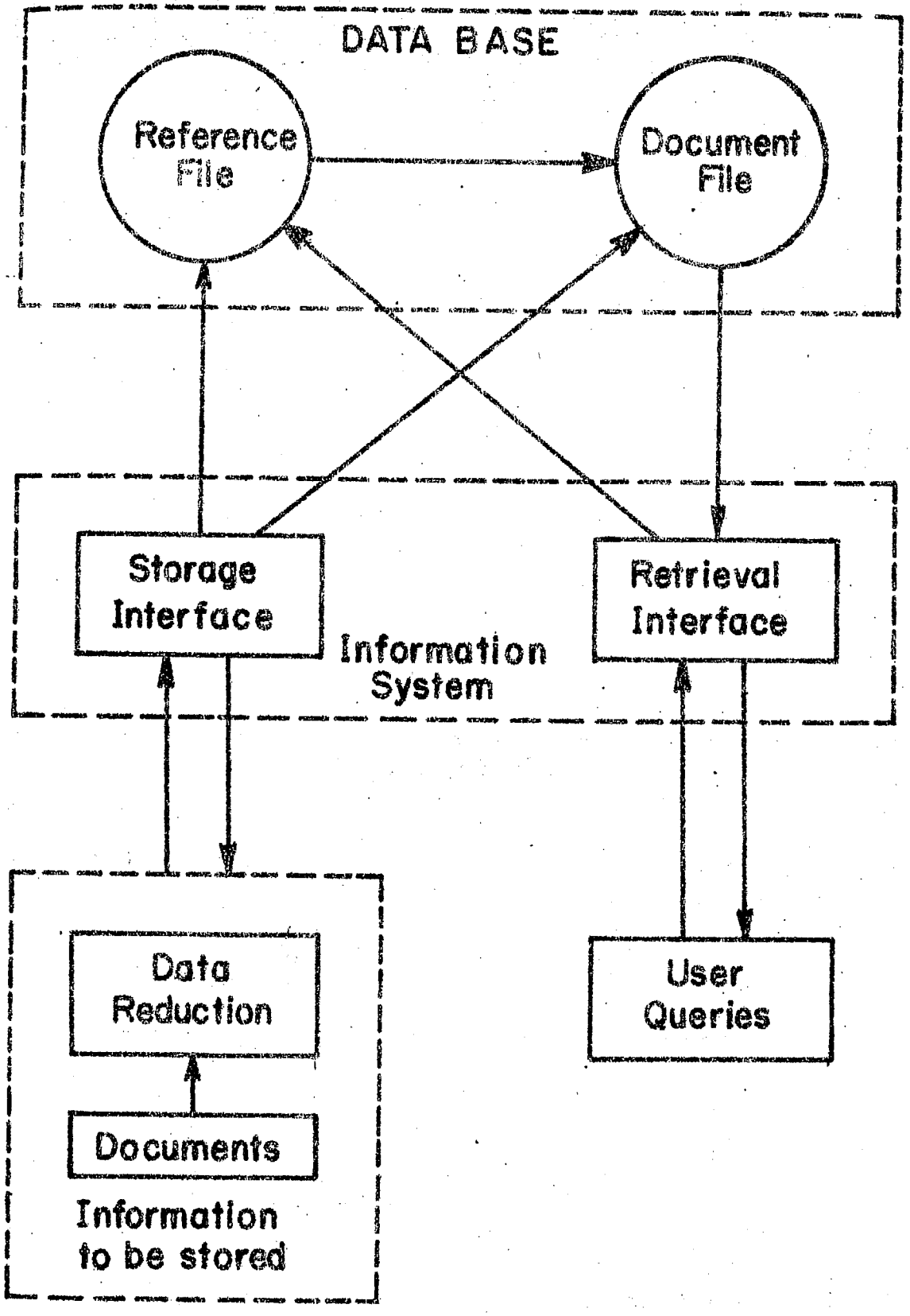
The computer based information systems (1-5) are ingeneral aimed at fulfilling two main tasks, namely (1) to maintain the information (i.e. to store and from time to time update the information on a suitable storage device) and (2) provide facility for reasonably fast retrieval of information on a request.

The storage of the information should be properly organized so as to make the retrieval process fast enough. The user interface, which would entertain requests in the form of user queries, should be simple and at the same time powerful enough to allow the user to retrieve the information precisely of his particular interest. Further more it is rather important that the user should also be able to make a quick global survey of the information data base by inferring about the key words for which corresponding documents are available so as to make successive guided requests and thus be able to access full amount of information of his interest. The retrieval interface should provide the facility to the user for various different modes of retrieval i.e. displaying on a terminal, printing out on a line printer or filing up on a storage disk file.

Rather than using a sequential file structure for storage of information, an appropriate Data Base Management System (DBMS) may be used which would allow for a fast and efficient retrieval

of information. The Data Bases imerged in the early 1960's with the underlying basic concept, namely to seperate the structure of the data from the program and thereby giving rise to data independence and the ability to set up data structures which are not strictly linear physical sequential file organisations. In 1969 CODASYL Data Base Task Group (DBTG) came up with a proposal[6] for the inclusion of data manipulation and data definition facilities into a host language, using COBOL as an example. A number of DBMS systems have then been built based on the DBTG specifications and incorporating other host languages like for example Fortran, Simula etc.

Various components of a typical computer based information system are shown in figure 1. The information system interacts with the data base during the process of storage of information and the retrieval of information based on user queries. The subcomponents of the information system which would help carrying out the above processes are shown in the figure as Storage Interface and Retrieval Interface. With reference to the figure the storage procedure would proceed as follows. The information to be stored in the data base is first collected and organized into different types of documents. For each document the information regarding a series of related attributes is disseminated according to the format compatible with the specifications of the storage interface. This process is known as data reductio Once the formatted information regarding the documents is obtain the storage interface is switched on. The interface would typic




A typical Information System

Fig. 1

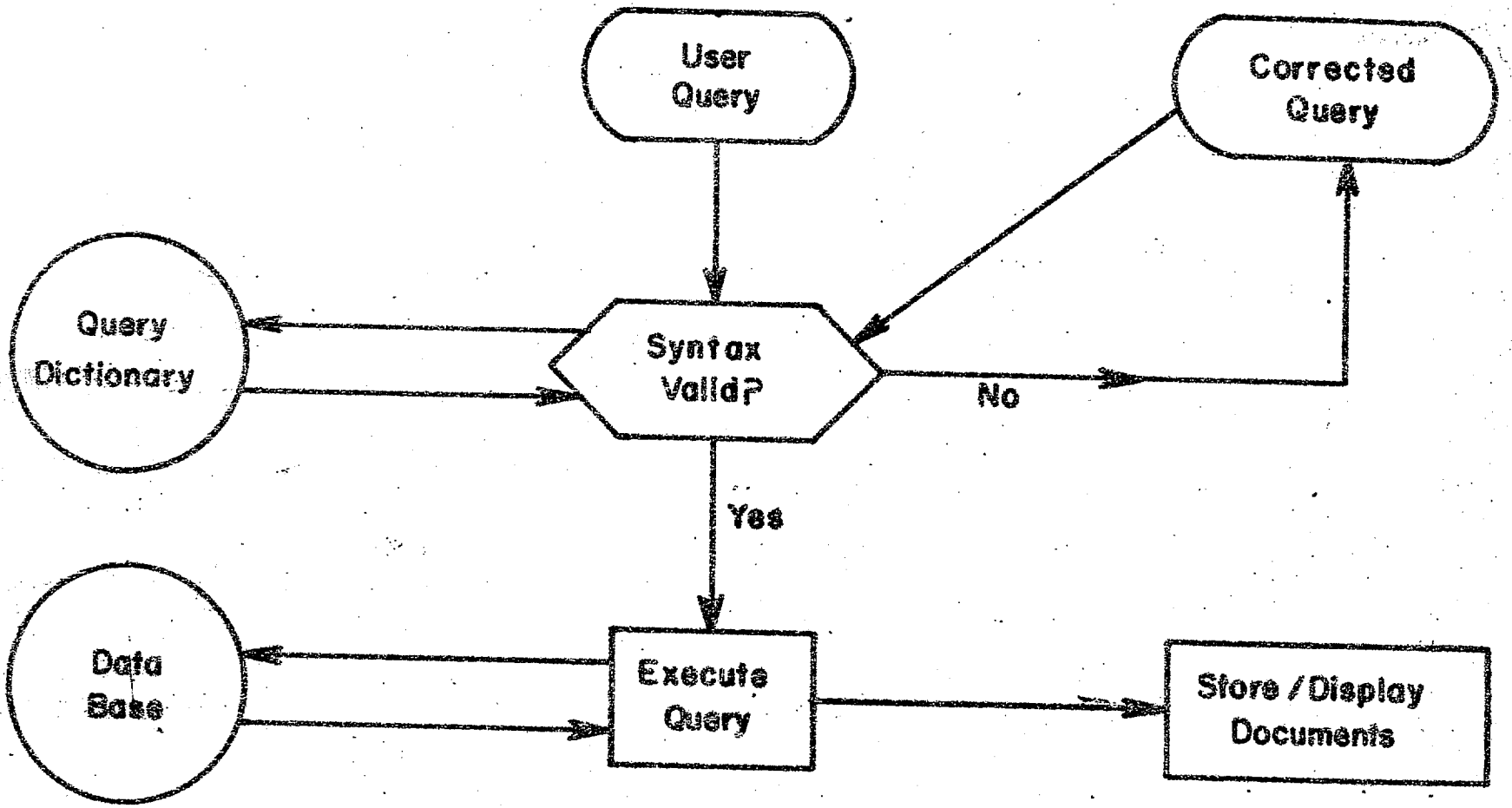
allow for the storage either in an interactive mode or in a batch mode. In an interactive mode the interface would successively give prompts for various information elements regarding the documents and accordingly the information may be interactively fed in. For a batch mode the formatted information required by the interface is first prearranged for a large number of documents on a disk file. The batch mode would then read the information from the file and accordingly pass it on to the data base.

As shown in the figure 1 the data base is typically made up of two parts, namely, a reference file and a document file. Whereas the reference file keeps proper indexing of the documents the actual information about the corresponding documents pointed to by the reference file is stored in the document file. Thus the reference file serves the purpose of a directory of certain keywords associated with and designating the documents stored in the document file. It is therefore possible to take resort to the directory and access the information regarding all the documents pertaining to given keyword(s).

Once the information data base is ready, the users would then be able to retrieve selective information by supplying queries to the retrieval interface.

b) Interactive Processing of User Queries

Every time a User comes up with a query the retrieval interface would first check for the validation of the query by means of a Query Processor (See Figure 2). If the Query words do not match with the once stored in the query dictionary




Processing User Queries

Fig. 2

or if the syntax of the query is not valid then the user would be informed so and asks the user to again supply the corrected version of the query. On getting a valid query the retrieval interface collects certain keywords from the query and passes them to the reference file of the data base. If the keywords are available in the reference file then the list of corresponding documents (pertaining to the given keywords) is obtained from the documents file and passed on to the retrieval interface. These documents are displayed to the user by the retrieval interface thus fulfilling the query command.

c) Some existing Information Systems

We mention here two information systems which are quite often referred to in the literaute, namely, 1. SMART System[5] and 2. SUPARS System[2]. Both are document retrieval systems and their salient features are as follows:

1. SMART System

The SMART System[5] has been developed by Gerard Salton and his group in PL/1 and FORTRAN IV languages. The system has been implemented at Cornell University, USA, on an IBM 7094 computer since 1964 and on IBM 360/65 computer since 1968. The system takes documents and search requests in English, performs a fully automatic content analysis of the texts, matches analyzed documents with analyzed search requests and retrieves those stored items believed to be most similar to the queries. The system has been tested with one or more of the following document

collections : a set of 780 abstracts of documents in computer literature, 1600 abstracts in aerodynamics, 1200 documents in documentation, 82 short papers on documentation and about 270 documents in medicine. The retrieval process is based on a concept of clustering for fast retrieval. However, the data base management system (DBMS) method is not resorted to for storage of information.

2. SUPARS System

The SUPARS (Syracuse University Psychology Abstracts Retrieval Service) System[2] is an online information retrieval system providing information in the form of abstracts of papers on Psychology. The query language supported by the system is rather simple in that documents related to a given subject name is made possible but the retrieval based on author name, publisher name, publication year etc. is not supported.

d) Introduction to LIBDBS

In the present note we consider the application of the information system for the usage of a Library in which the information to be processed is the bibliographic information for various documents, namely Books, Proceedings and Journal Article Considering the preliminary concepts of an information system discussed earlier a two fold role of the Library Data Base Syste (LIBDBS) is envisaged :

1) The Library staff would organize the information regard ing the documents available in the Library and Stores the

structures information onto the data base. A periodical updation of the data base from time to time (i.e. including new arrivals of documents and weedings out the outdated documents) would keep the data base upto date.

2) The Library users may make use of the data base (either directly or through the help of the Library staff) for quickly getting selective bibliographic information by supplying appropriate queries to LIBDBS. Some typical examples of the queries that LIBDBS would entertain are as follows:

Queries regarding documents:

* Give me a list of Books/Articles written by a particular author.
* List out the Books/Articles published in a given year.
* Display the list of Articles published in a given year (or by a given author).
* Which are the Books classified under a given classification number.
* Supply the abstract of a particular article appeared in a particular journal.
* List out all Books/Articles pertaining to a given subject keyword.

Queries regarding keywords (which help to form above queries)

* Are there any author keywords which start with ROS (or with Gri..... etc.)
* Give me a list of subject keywords starting with Comp (or with Cosmo.....etc.)
* List out all the publishers of Books/Proceedings.
* How many Books there are pertaining to a given subject keyword and published after a given year.

In Chapter 2 the design of LIBDBS system is described in detail. The User interface of the system describing the input commands to LIBDBS for the purpose of storage and retrieval of information from the data base is dealt with in Chapter 3. A brief summary follows through in Chapter 4.

# CHAPTER 2

## DESIGN AND COMPONENTS OF LIBDBS

This chapter describes in detail the integral components which make up LIBDBS and the system support it requires.

### a) The General Layout of the System

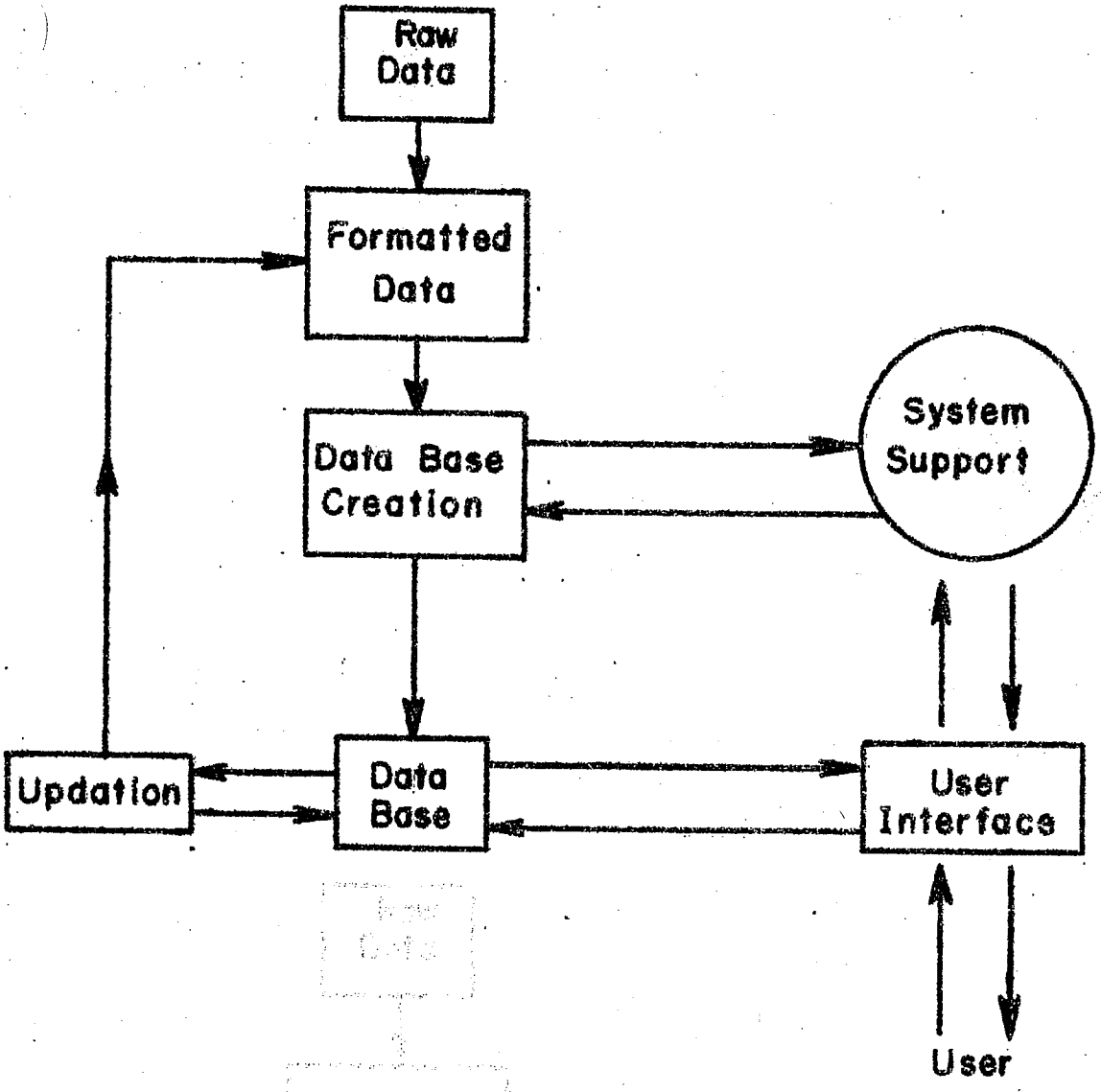
Various processes involved during the run time of LIBDBS and its general layout is shown in figure 3. It shows how the data base containing the structured bibliographic information is created and the selective retrieval of information is made possible. The successive steps involved during this process are described below (with regard to figure 3).

Step 1 : Data Base Creation:

In order to store structured bibliographic information in the data base it is first necessary to initialize the data base by storing the schema which contains the logical structure of various records related to different documents and their inter-relationship. Schema is stored in the data base file with the help of Simula's two system programs[11] namely, SPEC and PREP. A control file DBS.MIC for the purpose of initialization (i.e. storage of schema) of the data base file LIB.DBS is developed. The MIC file contains commands to run the programs SPEC and PREP as well as various input subcommands required by the two programs. Thus a single monitor level command

.DO DBS.MIC ⟨CR⟩

would do the job of initializing the data base file LIB.DBS.




Various Components of LIBIRS
(Library Information Retrieval System)

Fig. 3

This process is shown in figure 3 as 'Data Base Creation'. It requires system support for running the programs SPEC and PREP. The resulting file LIB.DBS thus created during the process is designated in the figure as 'Data Base'.

It may be noted that the initialization of the data base file need to be done only once - unless at a later stage the file is somehow damaged or lost. Once the data base file is initialized and stored with bibliographic information it may be later updated or used for retrieval purpose without further need for reinitialization.

Step 2 : Collection of Raw Data :

Having initialized the data base the information regarding various documents to be stored in the data base may be collected. The documents may be classified according to the type of the document. The collection of this information is shown in the figure as 'Raw Data'.

Step 3 : Organizing the Raw Data :

During a storage session with LIBDBS the documents (i.e. their related information) are stored one by one. For each document LIBDBS would give prompts for supplying various attributes in a fixed sequence (e.g. Accession number, followed by title followed by authors names etc.) Moreover each attribute has a fixed type (text, integer, real etc.). Thus it would be necessary to organize the raw information into the appropriate structured information depending on the type of each document.

If only a few documents need to be stored in a given session then the interactive storage mode may be selected in which case the organized information may be kept ready on a piece of paper. However, in case a large number of documents need to be stored in the data base in a given session then the interactive mode would be rather time consuming. In this case a batch mode of storage may be selected. Here the organised information may be stored on a disk file as a preparation for the storage in the data base file LIB.DBS.

The process of collecting the organized information is represented in the figure as 'Formatted Data'.

Step 4 : Storing the information in the Data Base :

Once the organized information about various documents is available it may be passed on to the system which would store it in the data base file LIB.DBS. The details of input command to be given to LIBDBS are given in Chapter 3.

Step 5 : Updation of Data Base :

From time to time the data base may be updated by storing the information about newly obtained documents as well as making desired changes in the stored information. The process of updating the data base, which is an important integral part of the data base maintenance, is shown in the figure as 'updation'.

Step 6 : The User Interface :

The data base thus created would be made available to the users who seek bibliographic information based on certain

keywords. The 'User Interface' as shown in figure 3 accepts queries from the user, processes them and accordingly makes available the desired information to the user in one of the three retrieval modes as specified by the user, namely displaying on a terminal, printing out on a line printer or storing on a disk file.

b) SIMULA and its DBMS support

LIBDBS is fully coded in Simula language[10-12] and is made up of about 3000 lines. Simula is ideally suited for the present application because of its excellent string processing facilities and the feature of the programming construct CLASS which helps in processing the records of the data base. Simula being a block structured language (like ALGOL) further helps a great deal in various system development stages, namely coding, debugging and addition of new features to the system as and when required. Another useful aspect about Simula is that it supports the data base management system (DBMS) packages SIMDBM and DBMSET[11] which are themselves entirely written in Simula. Whereas SIMDBM processes all operations regarding RECORD, the SET manipulations are carried out by DBMSET. The DBMS packages are based on CODASYL type[6] and are quite useful for the purpose of generation and updation of a data base of a moderate size.

Regarding the DBMS manipulation, use is made of two utility packages, namely, SPEC and PREP. SPEC is run to define records and sets. It generates random access data base file

in the form required by SIMOBM. PREP generates a file made up of CLASS declarations for all the records supplied previously to SPEC.

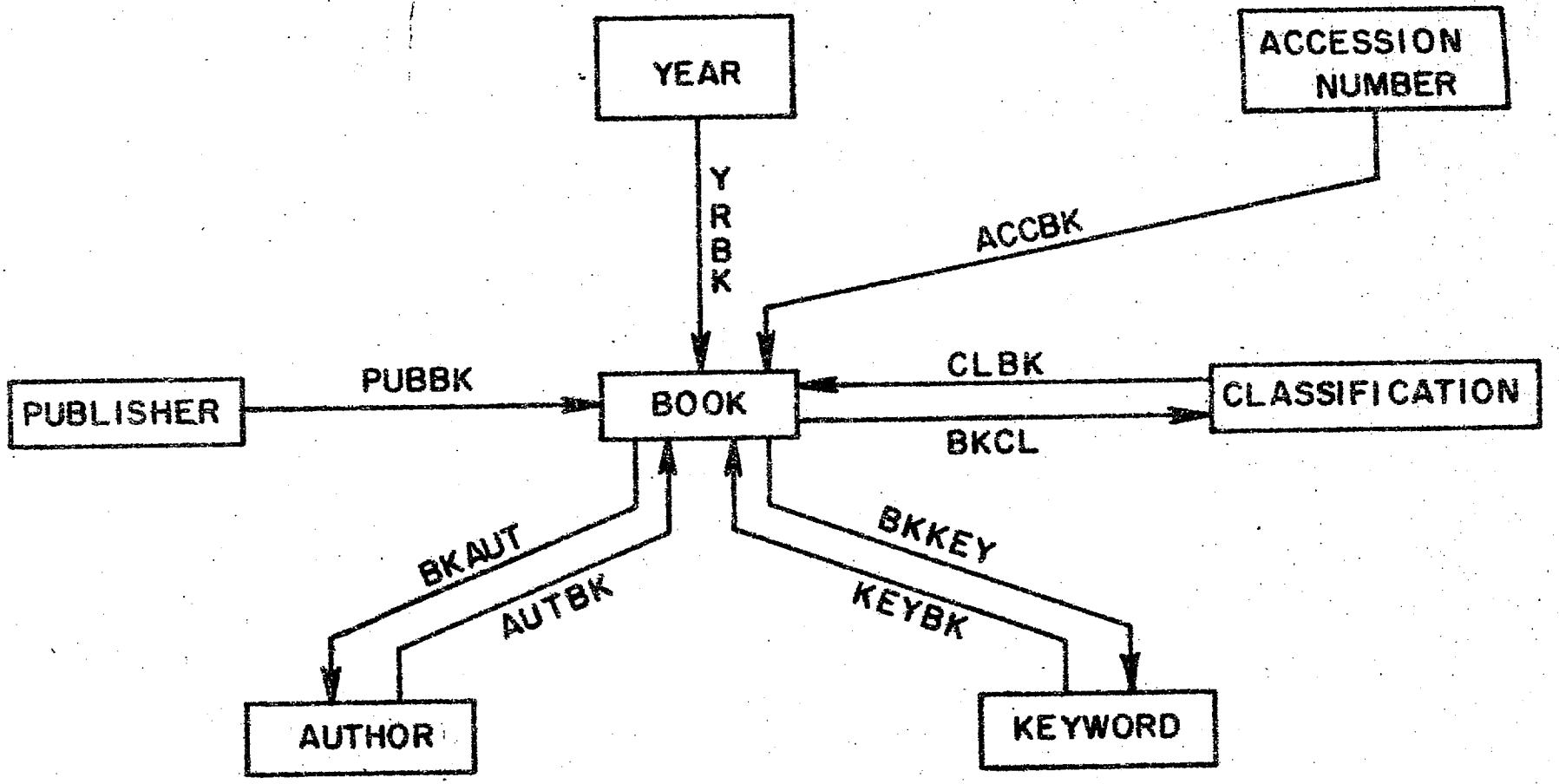
c) Outline of the Schema

A schema[6-9] describes an overall logical structure for the data base. It gives a complete description of various records, their associated attributes and the set occurences which define the possible linkages amongst the records.

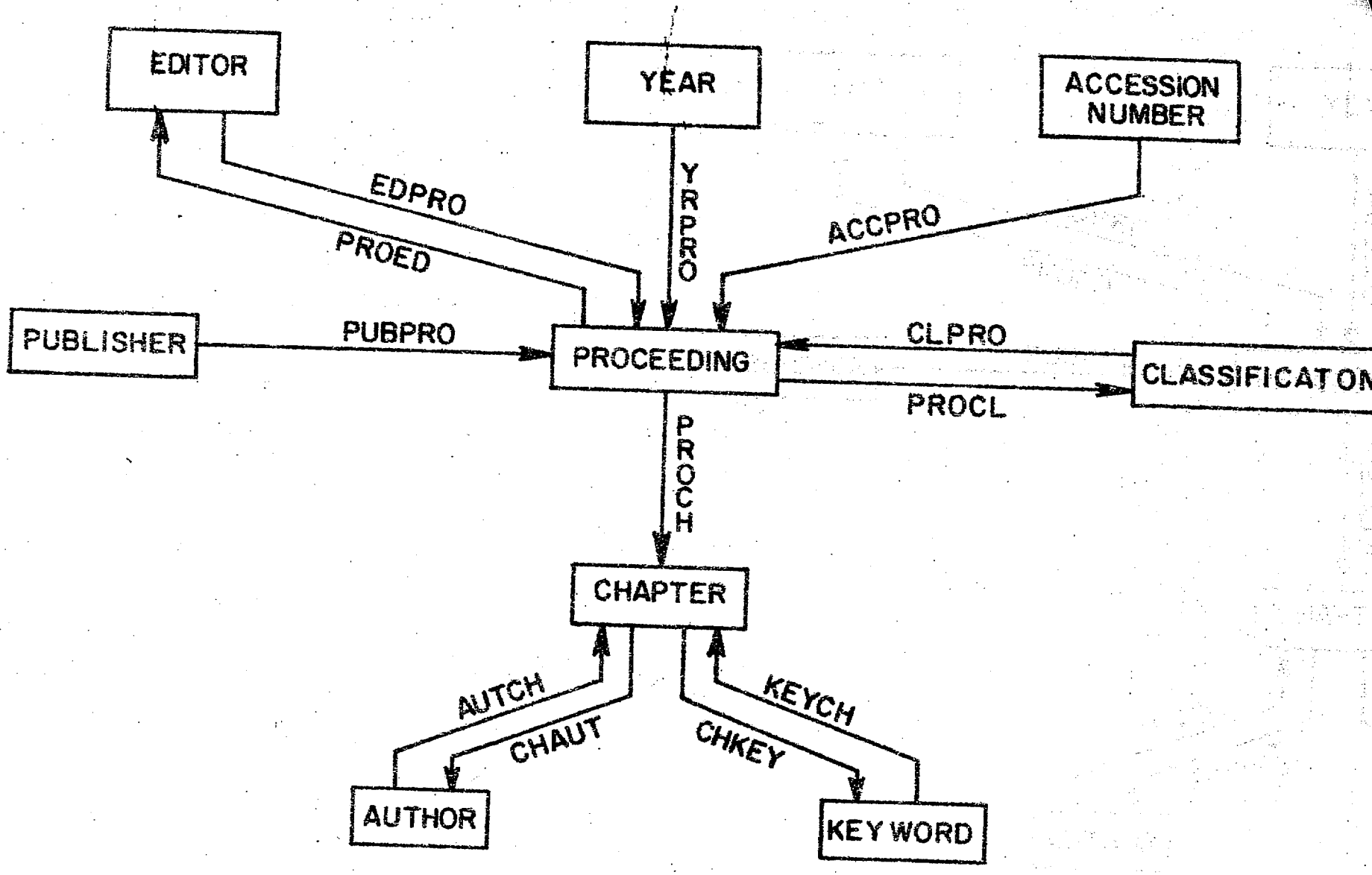
LIBDBS employs an overall schema for the data base. However for the purpose of clarity the schema is grouped into three subschemes according to the types of the Library documents, namely books, proceedings and journal articles. Figures 4, 5 and 6 describe the subschemas for these documents. The figures show the logical structure of the interrelationships of various records associated with the documents. As may be noted, many of the records are common to the three figures and individually they represent the same logical record in the schema of the data base. (The records which occur in more than one subschemas are Year, Publisher, Author, Keyword, Classification and Accession Number).

Appendix I contains more details about the schema, i.e. the fields and types of various attributes that make up various records and the set specifications.

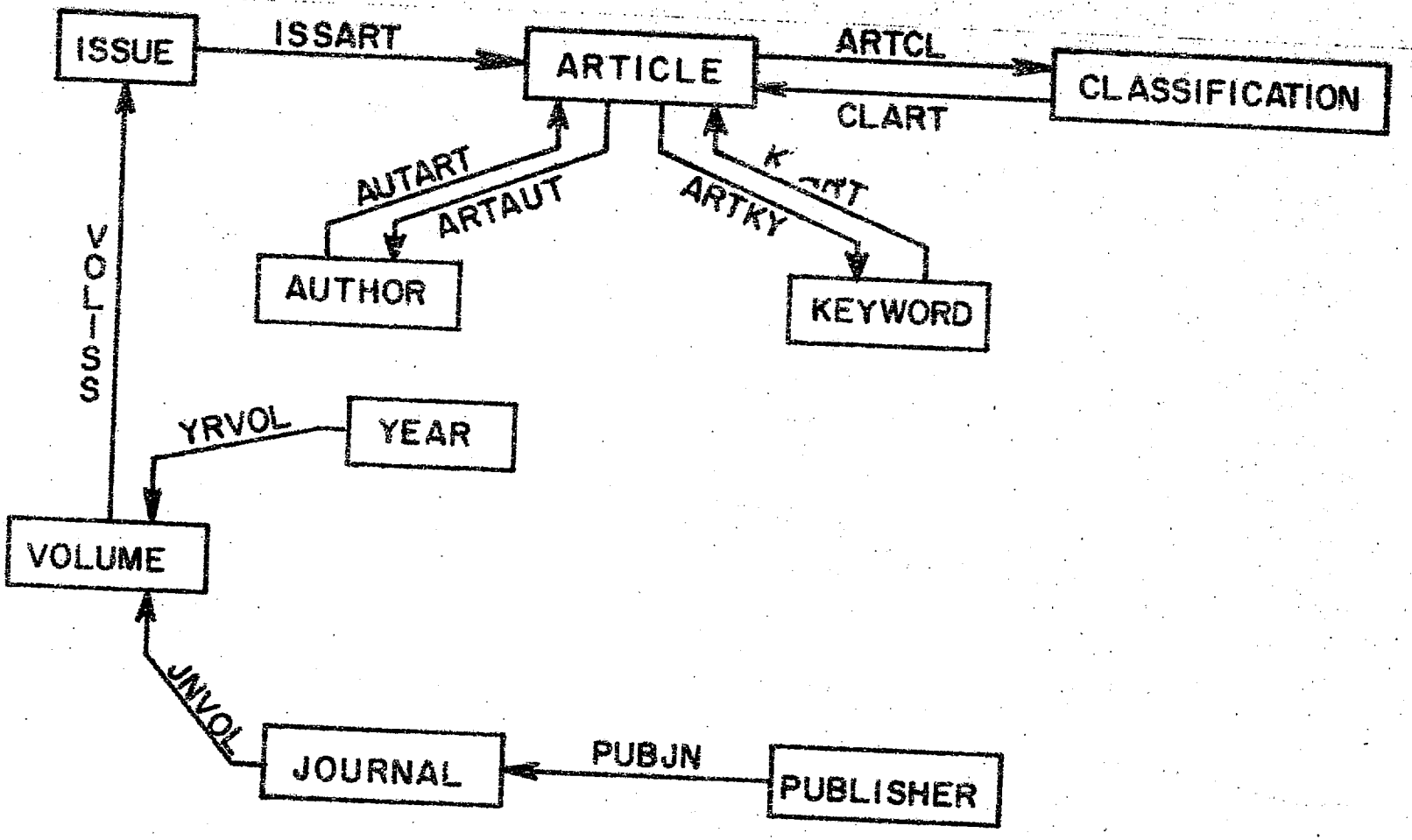
In order to appreciate the significance of the logical structures of the subschemas in figures 4, 5 and 6 consider the subschema for Books in figure 4. The record named BOOK




SCHEMA FOR BOOKS




SCHEMA FOR PROCEEDINGS




SCHEMA FOR JOURNAL ARTICLES

Fig.6

is the most significant record which contains full information about all the attributes of a Book in it, namely, Title, Authors' Affilliations, Publisher, Year of Publication, Number of pages, Price, Keywords, Accession Number and Classification Number. Thus once a BOOK record is located from the data base, all the relevent information regarding the book in the record is available and this may be right away displayed to the user. If the user queries were restricted to 'Give me a complete list of all the books in the library', then the subschema regarding Books would need to contain the only one record, namely, the most significant Book record. However, a user is seldom interested in a host of all kinds of books available in the Library. Rather the typical user queries concern with the list of books pertaining to a subject keyword or written by a particular author or published during a given year etc. (we shall use the term 'keyword' to represent one of the attributes, namely, subject keyword, author, publisher, publication year, Accession Number and Classification Number, based on which a list of documents is asked for in a query. Note that sometimes the word 'keyword' would be used to stand for 'subject keyword'.)

In order to easily locate from the data base relevent documents based on given keywords, it is therefore necessary to form seperate records representing each possible keyword and link them with the Book record. As shown in figure 4, different records representing specific attribute keywords are linked to the Book record via the appropriate set types.

For example the Year record is linked with the Book record by the set type YRBK.

Consider a user query which is intended to getting a list of all books written by the author N.Wirth. The data base would first be searched for all the Author records that have their Author name attribute as N.Wirth. Having obtained this list of relevent Author records it would be straight forward to obtain the associated list of all corresponding Book records through the set type YRBK. Since the Book records have all the relevent information regarding the books, the corresponding information may be made available to the user. This illustrates typically how all books based on a given keyword ('N.Wirth' in the above example) are retrieved from the data base. With reference to figure 4 it is thus clear that with the appropriate usage of the schema, LIBDBS would entertain queries which ask for a selective list of books written by an author, based on a subject keyword, published by a publisher, published in a given year, having a given accession number or having a given classification number.

Similar discussion holds for the schemas for PROCEEDINGS and ARTICLES. Referring to figure 5 it may be noted that it would be possible to retrieve a list of proceedings based on one of the 7 keywords, namely, Year, Editor, Publisher, Authors, Subject keywords, classification Number and Accession Number. And referring to figure 6 it may be noted that a list of journal articles may be retrieved based on one of the 8 keywords, namely Publisher, Journal name, Publication Year, Volume Number, Issue Number, Subject keywords or Classification Number.

d) Data Base Generation

Simula supports a CODASYL type DBMS system known as SIMDBM[11] which itself is entirely coded in Simula language. SIMDBM is quite useful for manipulating with data base of a moderate size. In the present application it is required to generate the data base filled with bibliographic information about the documents available in the Library. The process of data base generation involves two steps which are described below:

Step 1 : Storage of Schema in the Data Base File

The data base file must first be stored with the schema (described in the previous section) which would contain the overall logical structure of the data base. In particular this involves specifying various attributes and their corresponding types for each record and also specifying the sets which define how various records are linked with each other. This initialization of the data base is done using Simula's two utility programs, namely, SPEC and PREP. The program SPEC helps define specifications to the records. The input required for SPEC is similar to the Data Definition Language (DDL) in the CODASYL proposal[6]. SPEC generates an output file which is a random access data base file containing record specifications in a format that is needed by SIMDBM. The program SPEC is also used to make specifications for sets. The utility program PREP is used to generate a file having the CLASS declarations of all the records of the schema. This helps for an easy access

method for any field of a given record by supplying the attribute name representing the field and the record name.

A control file LIB.MIC is developed which contains commands to run the programs SPEC and PREP and the various inputs required by the two programs. Thus a single monitor command

.DO LIB.MIC ⟨CR⟩

would do the job of initializing the data base file by running the programs SPEC and PREP.

Step 2 : Storage of Bibliographic Information in the Data Base:

The storage of information about various documents is done by running the main program of LIBDBS, namely LIB.SIM. The details of various inputs to the commands given by LIB.SIM. during a storage mode (the other mode being the Retrieval mode) are given later in the Chapter 3 section (a).

One of the efficient ways to carry out a mass storage of information about a large number of documents in the data base file would be as follows. The batch mode of storage is most suitable for a mass storage. According to the types of documents, 3 seperate data files, namely, BOOK.DAT, PROC.DAT and ART.DAT may be prepared containing formatted data about Books, Proceedings and Journal Articles respectively. The program LIB.SIM would then be run switching on the batch mode of storage. Given one of the data file names, LIB.SIM would sequentially read a given data file and store the information about successive

documents in the data base. The data files may be supplied successively to LIB.SIM thus generating the data base file LIB.DBS stored with structured information about the documents. It may be noted that the data files may serve the purpose of a backup store for the data base. In case of a possible loss or damage to the data base file, the data files may be used to again create the data base file.

The interactive storage mode may be used to update the data base as and when required.

e) The Query Interpreter

Once the data base is generated it would be made available to the users seeking selective information regarding various documents. The users would interact with the data base via the Retrieval Interface (shown in figure 1) by supplying appropriate user queries. Typically the user queries are processed in 3 steps as described below (see also figure 2).

Step 1 : Validation of Query:

Every time a user comes up with a query the retrieval interface first checks for the validation of the syntax of the query. A certain set of allowed query words (about 50 words) makes up the vocabulory of a simple query language. The syntax of the language supports queries in the form of natural English language sentences like e.g. 'Get me all books by Dieskstra E.W or 'List the proceedings published in 1979' etc. The details about the vocabulary and the syntax of the query language are given in Appendix II.

As shown in figure 2 the query processor checks for the validation of the query with the use of a Query Dictionary. If the query is found to be invalid, the user would be informed so on the display terminal with the message 'Invalid Query' and is asked to supply the corrected version of the query. If the query is found to be valid then following further action is taken.

Step 2 : Executing a Valid Query:

The query processor would scan a valid query and extract out 1. the type of the document (i.e. Book, Proceeding or Journal Article) of user's interest and 2. the keywords based on which the list of document is required. Depending on the type of the query the data base is accordingly searched. If the query is of general type then a list of all the keywords having the given partial search word as the initial word is obtained. If the query is of specific type then a list of all documents (of the given type) which are related to the given query words is obtained.

Step 3 : Retrieval of Information :

The bibliographic information thus collected from the data base is passed on to the display procedure which, according to the user's interest, sends it to the display terminal, to the line printer or to the specified disk file for storage purpose.

# CHAPTER 3

## THE USER INTERFACE OF LIBDBS

This chapter describes how various inputs may be given interactively to the various prompts given by LIBDBS during its run time. In order to distinguish between what is typed by the user on the terminal keyboard and what prompts and information messages are displayed by LIBDBS, the former is shown in the upper case letters while the latter is shown in the lower case letters. For the purpose of clarity the storage and retrieval of bibliographic information are shown in the following in two different run time sessions.

### a) Storage session

Consider that a book (entitled "The Ascent of Man") is to be stored in the interactive mode and a book (entitled "The Voyages of Apollo") is to be stored in the batch mode. The storage session with LIBDBS may proceed as follows. ( ⟨CR⟩ represents the RETURN key on the terminal keyboard which gives a carriage return and line feed).

```
.RUN LIB ⟨CR⟩

          ************************************************
          libdbs : the library data base system
          ************************************************
data base file specification/lib.dbs/ :   ⟨CR⟩
Command : HELP  ⟨CR⟩
     give one of the following 3 commands :
```

update : to store/update the data base

retrieve : to retrieve information from data base

exit : to exit from libdbs

Command : U ⟨CR⟩

the update session starts.

updating interactively ? (Type y, n or e)/y/ : ⟨CR⟩

document name/book/ : ⟨CR⟩

enter attributes of the book

accession number ( e to exit) : 7731 ⟨CR⟩

title : THE ASCENT OF MAN ⟨CR⟩

authors : J. BRONOWSKI ⟨CR⟩

affiliation : (NOT KNOWN) ⟨CR⟩

keywords : HUMAN EVOLUTION, ASCENT OF MAN, POPULAR SCIENCE CR

price : Rs.220 (in 1982) ⟨CR⟩

Publishers : BRITISH BROADCASTING CORPORATION ⟨CR⟩

Year : 1973 ⟨CR⟩

Classification number : P 001/BRO/7731 ⟨CR⟩

Pages : 448 ⟨CR⟩

wait till the book is stored.

The book "The ascent of Man" stored in LIB.DBS

accession number (e to exit) : E ⟨CR⟩

Command : U ⟨CR⟩

the update session starts

updating interactively ? (Type y, n or e)/y/ : N ⟨CR⟩

data file specification : ABOOK.DAT ⟨CR⟩

document name/book/ : ⟨CR⟩

wait till books are stored.

The book "The Voyages of Apollo" stored in LIB.DBS

Command : E CR

During this storage session the two books are stored in the data base file LIB.DBS. The attributes of the first book are supplied interactively by typing out each attribute on the keyboard of the terminal. For the second book the attributes are not provided interactively. Rather they are first stored in the data base file ABOOK.DAT (not carried out during the above session) and only the file name is supplied interactively. The system reads all the attributes successively from the data file and stores the book in the data base.

The input sequence of various attributes for Book, Proceedings and Journal Article is given in Appendix III.

b) Retrieval Session

Consider that some few documents have been stored in the data base. A typical retrieval session which would supply various general and specific queries and get corresponding information would proceed as follows:

```
.RUN LIB   <CR>
        ********************************************
        libdbs : the library data base system
        ********************************************
data base file specification/lib.dbs/:        <CR>
command  :  R       CR
```

the retrieval session starts

* AUTH (BOOK) : CAL ⟨CR⟩

Calder N (2)

Caldwel S.W. (1)

* GET ALL BOOKS BY CALDER N. ⟨CR⟩

2 books :

Title : The key to the Universe

Authors : Calder N

Publication Year: 1977

Keywords : Universe, Popular Science, Big Bang

Title : The Life Game

Authors : Calder N

Publication Year: 1973

Keywords : Life, Popular Science

* KEYW(BOOK) : DO ⟨CR⟩

| | |
|---|---|
| double stars | (5) |
| double helix | (1) |
| documentation | (3) |
| dolphin | (2) |
| doorway state | (2) |

* GET THE BOOK (ALL) ON DOUBLE HELIX ⟨CR⟩

1 book :

Title : The Double Helix

Authors : Watson J.D.

Affiliation : Harward Univ., Cambridge

Publisher : Weidenfeld and Nicolson

Year : 1968

Keywords : double helix, dna, genetics

Price : Rs.105 (in 1979)

Accession Number : 9372

Classification number : P 575 : 612 : 01/WAT/9372

Pages : 226

*E ⟨CR⟩

Command : E ⟨CR⟩

•

The session shows how the general type of query helps in locating documents based on keywords (like author and subject keyword in the above session) and then retrieve the information regarding the documents by giving the specific queries. Note that the information obtained as a result of giving a generaly query contains the list of desired keywords and the corresponding number of documents available in the data base are shown within brackets. For example

doorway state (2)

informs the user that the data base contains 2 books related to the subject keyword 'doorway state'.

# CHAPTER 4

## SUMMARY

In the present note the design and the salient features of the Library Data Base System (LIBDBS) have been presented for its usage on the DEC-10 computer system. The note would also serve the purpose of the user manual for the system. To help get familiar with the working mode of the system a demonstration program is developed, the details of which are given in Appendix IV.

Two extensions to the system are envisaged : 1) The AND and OR constructs may be included in the query structure thus allowing for the retrieval of bibliographic information based on more than one keywords. 2) Besides the 3 types of documents considered in the present system, other documents, like for example Technical Reports, Reprints, Reference Books and PRL Publications etc. may be included.

REFERENCES


1. Senko, M.B., Altman, R.B., Satrhan, M.M. and Fehlder, P.L. "Data Structures and Accessing in Data Base Systems" IBM Systems Journal Vol.12, No.1, 30-93 (1978).

2. Lancaster, F.W. and Fayer, E.G. "Information Retrieval On-line", Melville Publishing Co. (1973).

3. Swihart, S.J. and Hefley, B.F. "Computer Systems in the Library", Melville Publishing Co., (1973).

4. Lefkovitz, D. "File Structures for On-line Systems", SPARTAN Books, New York (1969).

5. Salton Gerard "The SMART Retrieval System", Prentice Hall (1971).

6. CODASYL DBTG group "Data Base Task Group (DBTG) Report", April (1971).

7. Date C.J. "An Introduction to Data Base Systems", Addison Wiley (1975).

8. Widerhold, G. "Data Base Design", McGraw Hill, (1977)

9. Trembly, J.P. and Sorensen "An Introduction to Data Structures with Applications", McGraw Hill (1976).

10. Graham M., Birtwistle Dahl, O.J., Myhrhaug, B and Nygaard "Simula Begin", Petrocelli (1973).

11. SNDRI (Swedish National Defence Research Institute) "DEC System 10 Simula Language Handbook Part I, II and III", Digital Equipment Corporation (1972).

12. SNDRI "DEC System 10 Simula Technical Documentation", Digital Equipment Corporation (1972).

ACKNOWLEDGEMENTS

The author is grateful to Dr. S.Ramani and Mr. P. Sadanandan of NCSDCT, TIFR, Bombay for suggesting the outline of LIBDBS, for providing guidence and for many interesting discussion regarding its design. Thanks are due to Mr. R.Chandrasekhar (Visiting Member at NCSDCT) for many stimulating discussions on features of Simula language. The debugging and testing of the system were carried out on the DEC 1077 computer system at NCSDCT, TIFR, Bombay. The author is grateful to Professors A.Sen and K.H.Bhatt for the encouragement in the present work and for reading the note and making several suggestions for its improvement. The interest and the moral support provided by Professor S.P.Pandya is acknowledged with deep sense of gratitude.

# APPENDIX - I

## DETAILS ABOUT THE SCHEMA

a) List of 14 records and their specifications. (Records are given in the alphabetic order)

1. ACCESSION NUMBER :

Name of the Record : ACC (key : ACCESSNO)

| Type | Attributes |
|---|---|
| Text | ACCESSNO |

2. ARTICLE :

Name of the Record : ART (key : ARTKEY)

| Type | Attributes |
|---|---|
| Text | TITLE, ABST, AUTH, AFFIL, KEYW |
| Integer | NOOFREF, ARTKEY, AVFLAG, IPAGE, FPAGE, NOOFPAGES, YEAR |

3. AUTHOR :

Name of the Record : AUT (key : AUTHNAME)

| Type | Attributes |
|---|---|
| Text | AFFIL, AUTHNAME |

4. BOOK :

Name of the record : BK (key : ACCNO)

| Type | Attributes |
|---|---|
| Text | TITLE, PRICE, ACCNO, AUTH, CLNO, KEYW, PUBL, AFFIL |
| Integer | NOOFREF, NOOFPAGES, VOLNO, YEAR |

5. CHAPTER :

Name of the Record : CHP (key : CHAPKEY)

| Type | Attributes |
|---|---|
| Text | TITLE, PAGES, ABST |
| Integer | NOOFREF, CHAPKEY, CHAPNO |

6. CLASSIFICATION NUMBER :

Name of the Record : CL (key : CLASSNO)

| Type | Attributes |
|---|---|
| Text | CLASSNO |

7. EDITOR :

Name of the Record : ED (key : EDNAME)

| Type | Attributes |
|---|---|
| Text | AFFIL, EDNAME |

8. ISSUE :

Name of the Record : ISS (key : ISSKEY)

| Type | Attributes |
|---|---|
| Text | DATE |
| Integer | ISSKEY, AVFLAG, ISSNO |

9. JOURNAL :

Name of the Record : JN (key : JNLNAME)

| Type | Attributes |
|---|---|
| Text | JNLNAME, PERIODICITY, PRICE, AGENT |
| Integer | AVFLAG |

10. KEYWORD :

Name of the Record : KY (key : TOPIC)

| Type | Attributes |
|---|---|
| Text | TOPIC |

11. PROCEEDING :

Name of the Record : PRO (key : ACCNO)

| Type | Attributes |
|---|---|
| Text | TITLE, PRICE, VOLTITLE, ACCNO, EDIT, AFFIL, PUBL, CLNO |
| Integer | NOOFPAGES, NOOFCHAP, VOLNO, YEAR |

12. PUBLISHER :

Name of the Record : PUB (key : PUBLNAME)

| Type | Attributes |
|---|---|
| Text | PUBLNAME, ADDR |

13. VOLUME :

Name of the Record : VOL (key : VOLKEY)

| Type | Attributes |
|---|---|
| Text | ACCNO |
| Integer | VOLKEY, AVFLAG, VOLNO |

14. YEAR :

Name of the Record : YR (key : PUBLYEAR)

| Type | Attributes |
|---|---|
| Integer | AVFLAG, PUBLYEAR |

b) Sets (the linkage amongst records) and their specifications (The sets are shown in the alphabetic order)

| No. | Set Name | Owner Record | Member Record |
|---|---|---|---|
| 1 | ACCBK | ACC (Accession Number) | BK (Book) |
| 2 | ACCPRO | ACC (Accession Number) | PRO (Proceeding) |
| 3 | ARTAUTH | ART (Article) | AUT (Author) |
| 4 | ARTCL | ART (Article) | CL (Classification Numbe |
| 5 | ARTKY | ART (Article) | KY (Keyword) |
| 6. | AUTART | AUT (Author | ART (Article) |
| 7 | AUTBK | AUT (Author) | BK (Book) |
| 8. | AUTCHP | AUT (Author) | CHP (Chapter) |
| 9 | BKAUT | BK (Book) | AUT (Author) |
| 10. | BKCL | BK (Book) | CL (Classification Number |
| 11 | BKKEY | BK (Book) | KY (Keyword) |
| 12 | CHAUT | CHP (Chapter | AUT (Author) |
| 13 | CHKEY | CHP (Chapter | KY (keyword) |
| 14 | CLART | CL (Classification Number) | ART (Article) |

| No. | Set Name | Owner Record | Member Record |
|---|---|---|---|
| 15 | CLBK | CL (Classification Number) | BK (Book) |
| 16 | CLPRO | CL (Classification Number) | PRO (Proceeding) |
| 17 | EDPRO | ED (Editor) | PRO (Proceeding) |
| 18 | ISSART | ISS (Issue) | ART (Article) |
| 19 | JNVOL | JN (Journal) | VOL (Volume) |
| 20 | KEYBK | KY (Keyword) | BK (Book) |
| 21 | KEYCH | KY (Keyword) | CHP (Chapter) |
| 22 | KYART | KY (Keyword) | ART (Article) |
| 23 | PROCH | PRO (Proceeding) | CHP (Chapter) |
| 24 | PROCL | PRO (Proceeding) | CL (Classification Number) |
| 25 | PROED | PRO (Proceedings) | ED (Editor) |
| 26 | PUBBK | PUB (Publisher) | BK (Book) |
| 27 | PUBJN | PUB (Publisher) | JN (Journal) |
| 28 | PUBPRO | PUB (Publisher) | PRO (Proceeding) |
| 29 | VOLISS | VOL (Volume) | ISS (Issue) |
| 30 | YRBK | YR (Year) | BK (Book) |
| 31 | YRPRO | YR (Year) | PRO (Proceeding) |
| 32 | YRVOL | YR (Year) | VOL (Volume) |

# APPENDIX II

## VOCABULORY AND STRUCTURE OF QUERIES

Valid user queries are made up of certain keywords that make up the vocabulary (or edictionary) of the query language. The query may be formed from these query words according to a grammar, to be described shortly, which defines the syntax of the query. There are fixed number of parts in the two different types of queries, namely 6 for the specific query (which retrieves a list of documents based on given keywords) and 4 for the general query (which retrieves a list of keywords based on a given partial search word).

The general format of the two types of queries is as follows:

a) Specific Query :

A Specific query has 6 parts in the following order :

| No. | Part | No. of available keywords |
|---|---|---|
| 1 | Command Word | 6 |
| 2 | Free Words (optional) | 3 |
| 3 | Document Name | 3 |
| 4 | Arguments to Document (in round brackets) | 10, 10, 8 for Book Proceeding and Article respective |
| 5 | Condition | 8 |
| 6 | Keyword | (User given) |

Except the 6th part, i.e. keyword, which is given by the author the rest of the parts must contain words from the vocabulary of the query language. The available query words for the first 5 parts are as follows:

Part 1 — Command Word — Comment

DISPLAY
FIND
GET
LIST
OBTAIN
TYPE

(Note : All the command words have semantically the same meaning)

Part 2 — Free Words (Optional)

ALL
ME
THE

(The words are optional. More than 1 word may also be used in any orde

Part 3 — Document Name

ARTICLES
BOOKS
PROCEEDINGS

(One document name may be given)

Part 4 — Arguments to Document (Optional)

For book : TITLE
AUTH
YEAR
KEYWORDS

AFFIL

PUBL

NOOFPAGES

PRICE

ACCNO

CLNO

ALL

(This part is optional. One or more argument words may be given seperated by commas. Default arguments are first 4)

For Proceedings:

TITLE

EDIT

YEAR

KEYWORDS (Same as above (for books))

AFFIL

PUBL

NOOFPAGES

PRICE

ACCNO

CLNO

ALL

For Article:

TITLE

AUTH

YEAR (Same as above (for books))

KEYWORDS

AFFIL

NOOFPAGES

ABST

ALL

| Part 5 : Condition | Part 6 : Keyword |
|---|---|
| BY | Author |
| ON | Subject keyword |
| PUBLISHED BY | Publisher |
| PUBLISHED IN | Year |
| PUBLISHED SINCE | Year |
| PUBLISHED BEFORE | Year |
| CLASSIFIED UNDER | Classification Number |
| ACCESSED IN | Accession Number |

Examples of Valid Queries:

Get the Books by Bronowski J. ⟨CR⟩

Find me Books (all) on Nuclear Structure ⟨CR⟩

List the Articles on Tokomak ⟨CR⟩

Display Proceedings published in 1978 ⟨CR⟩

Get Books published by Petrocelli ⟨CR⟩

Ge B acc in 725 ⟨CR⟩

Ge B (tit, Auth) class und 533.9 ⟨CR⟩

Examples of Invalid Queries

Get Books all on Geocosmology (all should be within round brackets)

Books by McCarthy (first part is missing)

Find documents published by McGraw Hill ('documents' is not valid document name

Display books and proc on UFO (only one document is allowed)

b) General Query:

A general query is made up of 4 parts in the following order. Note that first two parts form a valid query and other two parts may be given optionally if necessary.

| No. | Part | No. of available keywords |
|---|---|---|
| 1 | Keyword | 6 |
| 2 | Document Name | 3 |
| 3 | Search Word | (User given) |
| 4 | Maximum Number | (User given) |

The available keywords for the first two parts are as follows:

Part 1 : Keyword :

ACCESSION

AUTHOR

CLASSIFICATION

KEYWORD (the subject keyword)

PUBLISHER

YEAR

Part 2 : Document Name :

ARTICLE

BOOK

PROCEEDING

Part 3 : Search Word:

: Search Word (The search word is to be given followed by : )

Part 4 : Maximum Number:

, Number (The maximum number is to be given followed by ,)

Examples of Valid Queries:

Auth (Book) : Sh, 5

Auth (Book) : Zim

Year (Pro)

Keyw (Art) : Fusi

Publi (Book) : Mc

Keyword (Book) : Comp, 10

Example of Invalid Queries:

Book (Auth) : K

(Mismatch of 1st two parts)

Publ : Proc.

(Proc should be within brackets rather than followed by :)

Auth (Journal) : Einst

(Journal is not a valid document name)

KeyW(Book) : Geo 10

(A, is missing between Geo and 10)

Note:

A general query is useful in the following sense. It would be good to avoid getting the message on the display terminal "sorry ! Information not available" in response to specific queries (like e.g. Get all books by Zenor) given on a trial and error basis. A user is usually not aware of the scope of the data base information. It is therefore recommended to first make use of a general query to get an idea of the keywords based on which information is available in the given library data base. Appropriate search words may be provided to get the list of relevent keywords of user's interest.

## APPENDIX III

The input sequence of various attributes (and their type) for Book, Proceeding and Article.

1) For Books :

| Type | Attribute |
|---|---|
| Text | Accession Number |
| Text | Title |
| Text | Authors |
| Text | Authors' Affilliations |
| Text | Keywords |
| Text | Price |
| Text | Publishers |
| Integer | Publication Year |
| Text | Classification Number |
| Integer | Number of pages |

2) For Proceedings:

| | |
|---|---|
| Text | Accession Number |
| Text | Title |
| Text | Editors |
| Text | Editors' Affiliations |
| Text | Keywords |
| Text | Price |
| Text | Publisher |
| Integer | Publication Year |
| Text | Classification Number |
| Integer | Number of pages |

3) For Journal Articles:

| Type | Attribute |
|---|---|
| Text | Journal Name |
| Text | Journal's periodicity |
| Integer | Publication Year |
| Text | Price of Journal |
| Integer | Availability Flag |
| Text | Issue Number |
| Text | Title of Article |
| Text | Abstract |
| Text | Authors |
| Text | Authors' Affilliations |
| Text | Keywords |
| Integer | Number of References |
| Integer | Initial Page Number |
| Integer | Final Page Number |

# APPENDIX IV

## THE DEMONSTRATION PROGRAM FOR LIBDBS

A control file LIBDEM.MIC has been developed to demonstrate the salient features of LIBDBS and to illustrate which successive input commands may be given at various stages during the run time of the system.

* How to run the demonstration program ?

The following two monitor level commands may be given to access the demonstration on LIBDBS.

. DO LIBDBS.MIC ⟨CR⟩

. DO LIBDEM.MIC ⟨CR⟩

It may be noted that the first monitor command initializes the data base with appropriate schema in the data base file TEST.DBS. The second monitor command then runs the main program LIB.SIM of LIBDBS and gives appropriate commands in response to prompts given by LIBDBS. Thus the first command is a pre-requisite to run the demonstration.

*What is demonstrated during the demonstration ?

The various actions taken during the demonstration are as follows:

Step 1 :-

The main program LIB.SIM is executed by giving the command .RUN LIB ⟨CR⟩

Step 2 :-

The data base file name TEST.DBS is supplied. At this stage the file does not contain information about any documents.

Step 3 :-

The UPDATE command is given to store the information about some books. The interactive mode of storage is selected. (The other storage mode being the batch mode which would be used later on).

4 Books are stored within the data base file TEST.DBS by supplying all the necessary attributes of each book in succession.

Step 4 :-

The RETRIEVE command is given to selectively obtain information about the 4 books stored in the data base.

Step 5 :-

Some general as well as specific types of queries are given and each time the corresponding information as asked by the query is displayed on the terminal.

Step 6 :-

Again the UPDATE command is given. This time the batch mode for the storage is selected. 3 more books are stored in the data base by reading the necessary information about books from a pregenerated data file named BOOK.DAT.

Step 7 :-

The RETRIEVE command is given to get selective information about total 7 books stored in the data base.

Step 8 :-

Some queries are given to retrieve information. The corresponding information is retrieved not on the display terminal but on a disk file named BOOK.LST.

Step 9 :-

The EXIT command is given to end execution of LIB.SIM program.

This ends the demonstration session.