PRL

TECHNICAL NOTE

TN-85-41

'READFAST' an online Information Retrieval System

By

D.R.Kulkarni and R.R.Bharucha

February,1985

Physical Research Laboratory
Ahmedabad-380009

CONTENTS

'READFAST' an online Information Retrieval System

D.R.Kulkarni and R.R. Bharucha

Abstract

'READFAST' system is a data-base management system developed for the library of the Physical Research Laboratory (referred to as PRL) Ahmedabad. The acronym stands for 'REtrieval of Acquired Documents through Friendly And STructured System'. The data-base consists of collection of acquired documents namely all books, scientific and technical reports received from other institutes and all the various publications of PRL. The system enables the user to retrieve the relevant documents using appropriate keywords, author-names, unique accession numbers, etc. The system is friendly as it helps the user to know the exact keywords or author-names available in the data-base which are nearest to ones provided by him. The pre-determined queries can be put to the system in a very flexible English-like sentence using a few reserved words. The system can guide the user to find the suitable keywords which are most relevant to the topic of his interest. This has been achieved by built-in hierarchy of keywords available in the data-base. The system also indicates the issue-category and the availability of the document in the library along with other useful details such as issue-status, date of issue and the library member to whom it is issued etc.

The system is based on the package DEMS-10 on DEC 1091. It has been fully implemented and tested on the simple

I. Introduction

To-day information is being generated in great volumes as it tends to cover broader and broader range of human activities and also tries to collect more and more current in-depth knowledge in each field. Thus the information has both breadth in terms of subjects covered and depth in terms of treatment given to subjects. It is, therefore, not surprising to see that the present pace, intensity and variety of human activities have given rise to what is popularly termed as 'Information explosion'. Usually the information that is generated at different sources is being documented and is made available to others through various well-established channels such as books, reports, journals, conference proceedings etc. As these documents are currently being produced with incredibly large pace, it has become necessary to evolve a system that not only collects, compiles and classifies the documents properly but should also be able to retrieve a particular document quickly whenever it is asked for. This is especially true in the fields of science and technology. In fact the accelerating rate at which scientific and technical knowledge is growing has created a pressing need for storing such information so that any publication or part of it would be available immediately when required. From user studies it has been found that a

scientist or engineer spends about 20-25% of his working time searching for information. Several published data are available which show the huge amount of money spent in duplication of research. Hence an efficient Information Retrieval System would ensure the maximum use of scientific and technical talent by reducing the time spent in locating information and would also preclude duplication of research.

II. Conventional retrieval in a library

Before we describe fully our on-line retrieval system, we would describe the conventional procedure adopted in a library to retrieve a document.

In general a bare document is qualified by the following attributes :

a. Title b. Author names c. Publisher name d. Publication year. Some of these attributes can be used to retrieve a document. However a document acquired by the Library undergoes some local processing through which it acquires one more attribute viz call number. A call number consists of classification number and an accession number. A classification number is assigned to a document based broadly on its subject and contents. An accession number is a unique attribute of the document and indicates the cumulative sequence number of the document in the library. The call number apparently helps the process of retrieval by classifying the document under different subject heads. It also helps in locating the document physically in the library. A typical library procedure

of retrieving a document involves searching various catalogues manually which generally provide access to a document either through title, author-name, or classification number. Though the access through title or author-name is relatively straight forward, in practice, its utility is limited due to various other factors. Very often a user may not remember the title of the document exactly. Though he may know the surname of the author, usually he may not remember his initials. The correct spelling of the surname is another bottleneck that one often experiences. If the author-catalogue contains details of only the first 2 authors then retrieval is not possible from the remaining author names. The retrieval through classification number which is based on the contents of the document seems to be attractive from the user's point of view. However, here also he faces even more problems. The classification number, in practice, can give the details of the contents only to a very limited extent. Consequently it is common experience that unless many documents are gone through, one is not likely to get a document which may be the most relevant to the topic of one's interest. In other words, the retrival through classification number produces more noise and hence gives less precision. The most important drawback of classification number is, however, that this number, being coded by the library staff following a standard classification scheme, does not signify anything to a

common user. Therefore, he cannot effectively use these numbers for retrieval without consulting the library staff, or the library catalog.

It is undoubtedly clear that retrieving a document by specifying in depth the contents of the document is the most extremely useful mode of retrieval as far as the user is concerned. The conventional library procedure, besides being slow, fails considerably in this aspect, as, even if 2 to 3 classification numbers are given to a book they do not reveal all the detailed subjects covered in the book.

III. Concept of keyword

Realising the shortcomings of a classification scheme, it was thought necessary that a document acquired by the library should undergo another phase of processing to get additional attributes that is indepth indexing. These attributes are the keywords. A keyword is a natural language word or phrase identifying uniquely a unit concept in a subject. The main advantage of keywords is that, being phrased in natural language, they can be understood both by the library staff as well as common users. Further it has been experienced that a document can be described better and with more accuracy by assigning number of keywords to it. Since the keywords represent the conceptual development of a subject, they form a vertical hierarchy showing intrinsic

semantic relationship between its terms. This hierarchy is a tree structure where top terms get split into number of narrower terms. The hierarchial relationship is extremely important as the user is led from broad subject (the top term) to more detailed specific subjects i.e. the narrow terms. It also improves the recall-precision ratio of the system by taking care of the terminological problems.

We use these keywords and their hierarchial relationships to retrieve documents in our system. A document has been assigned number of keywords derived from its title and contents. Before the keywords are assigned they are checked with our authority list and various thesauri in order to control the indexing terminology. Thus thesauri are used as tools for indexing. However, as we do not have thesauri for all the subjects in which our scientists are working, in several cases we have had to develop hierarchial relationships between various keywords of the subjects using the classification schedule and with the help of the subject specialists.

Besides the subject keywords, we have also coined some other keywords which reveal the level of treatment given to the subjects in different documents. These keywords can indicate if the document is an introductory level, or a textbook, a handbook, a conference proceedings, or if it is an historical presentation, a subject bibliography etc. These additional keywords help tremendously to retrieve the most relevant documents, thereby increasing the recall-precision ratio considerably.

In our system, retrieval is done using keywords and as retrieval is required at any level of hierarchy, we have to assign the complete hierarchy of keywords to each document for quick access. For example if a book deals with 'solitons', the following hierarchy of keywords may be assigned to it. (The numbers on the left denote levels).

0 Mathematics

1 Differential equations

2 Partial differential equations

3 Non-linear partial differential equations

4 Hyperbolic equations

5 Solitons

In short the hierarchy serves two purposes. The first one is that the book can be retrieved using any keyword in the hierarchy. Secondly it depicts the relationships between the keywords which is quite valuable to a novice in the subject.

IV. Objectives of READFAST system

READFAST system is an on-line, user-friendly information retrieval system based on the data base management package DBMS-10 available on system DEC-1091. The system has been developed for the library of Physical Research Laboratory. The acronym READFAST stands for REtrieval of Acquired Documents using a Friendly And STructured system. The data base consists of acquired documents in the library i.e. all books, scientific and

technical reports received from various organizations and also includes various PRL publications. These documents currently number approximately 20,000 and are likely to reach 30,000 in the next five years.

The design objectives of the system READFAST are to help the user in retrieving the documents in several ways.

1. The system READFAST basically provides three different access paths to a document using either keyword, author-name or accession number.

2. While retrieving the document using keyword or author-name, the user may not know keywords or author-names available in the data-base or their exact spellings. The system helps the user in this case by providing the list of keywords or author-names nearest to one he has given as input. There are also various other ways which will enable him to know the exact terms existing in the data-base.

3. A novice who does not know various sub-topics or the keywords associated with them in a particular subject can be helped by the system. He gets the guidance from the system through its built-in hierarchy of keywords so that he can obtain complete range of topics from broad to narrow terms. Thus the system caters to various types of users, those who are familiar with the subject and those who are new to the subject. This aspect is also quite useful to those who are not familiar with our data-base.

4. The system can also answer the complex queries which combine various keywords, author-names and the publication year or its range by binary operations.

5. The system will help the user in not only retrieving the document but also in providing the information regarding its availability in the library at a particular instant of time.

6. The system will allow various updates that are generally necessary in the course of time.

7. The system will give certain reports to help the library staff for the purpose of house-keeping.

In general the design of the system is tailored to answer the frequently - asked queries most efficiently. We discuss in the next section the contents and the design consideration of our system.

V. Design of READFAST data-base

In this section we briefly discuss some of the salient features of READFAST data-base. As mentioned before READFAST data-base consists of various documents acquired by the library. A document is characterized by four essential records associated with it. These four records are (a) Title-record, (b) Author-record, (c) Keyword-record and (d) Publisher-record. Besides these records, the data-base also contains various other records which make the system more efficient and user-friendly. We discuss below each type of record and its significance in the design of the data-base.

1. Title record :

This record essentially consists of title and accession number of the document along with other information such as publisher-code, year of publication, price, issue status etc. We describe below each field briefly.

i) Accession Number : As stated earlier, this is the unique attribute of the document in the Library. In our system, it consists of two parts. A single letter alpha code denotes if the document is a book (B), a report (R) or a PRL publication (P). The numeric-code is a usual 5-digit accession number. This implies that the system can have maximum 99999 documents in its data-base.

ii) Title : This field is meant for title to be given in maximum 125 characters. Besides the title, this field usually contains additional information such as edition number, details of the series or proceedings etc.

iii) Publisher-code : This is an 8 character code consistir of 2-digit numeric code denoting country of the publisher and 6-digit alphabetic code representing the publisher-name in abbreviated form.

iv) Year of publication : This is a 4-digit number indicating the year of publication of the document.

v) Price : This field stores the price of the document in rupees. This is a 8-character field.

vi) Call number : This is the call number of the document consisting of classification number as well as accession number and author code combined together. This has maximum 36 characters.

vii) Issue-category : This is a one-character alphabetic field that indicates if the document can be issued to a user. In general a document may be for general circulation (C), or for reference (R) or for restricted issue (O).

viii) Issue-status : This one-character alphabetic code shows the availability of the document in the library. The issue-status 'N' shows that the document has not been issued and hence available in the Library currently. The status 'I' shows that the document has been issued to a member. The code 'B' indicates that the document has been sent for binding. The codes 'L' and 'X' indicate that the document has been issued on a long term loan or inter-library loan respectively.

ix) Membership number: This field is a 3-digit number giving the membership number of a person to whom the document has been issued.

x) Date of issue : This is a 6-character numeric field consisting of date, month and year each of 2 digits. This is the date on which the document has been issued or sent for binding.

The accession number has been used as a calc key both for storing and retrieving the title record in the data-base. Since the accession number is unique, more than one document cannot have the same calc key.

2. Author record :

This record consists of single author-name with 17 characters for surname, 3 characters for maximum 3 initials and one character A or E to indicate if he is an author or editor respectively. For each author there would be separate record. The surname of the author has been used as a calc key for storing and retrieving the author-record. As same surname can be borne by more than one authors, same calc key is used for all of them. The author-name is fed in upper-case letters.

3. Keyword record :

This is a 45-character field consisting of upper-case alphabetic letters along with a under-score character to separate different words in the keyword. There will be a separate record for each keyword assigned to a document. Complete keyword has been used as a calc key to store and retrieve the keyword record. Obviously two keywords will not have the same calc key.

4. Alpha record :

This record is automatically generated by the system. The record essentially consists of two characters.

As new keyword and author record are inserted in the system, their first two letters are taken out to form a new alpha record. Subsequently all the author-names and keywords starting with the same two letters will be connected to this alpha record. The two characters in the alpha record form a calc key to store and retrieve these records.

5. Link 1 record :

The Link 1 record is generated by the system to connect title and keyword records together. They cannot be connected directly as they have many-to-many relationship which is not allowed in the network type data-base package. Instead a Link 1 record is created which splits this many-to-many relationship into two one-to-many relationships. The contents of Link 1 record are two data-base keys corresponding to a title record and the keyword record assigned to it. The access to this record is either through title record or keyword record.

6. Link 2 record :

Like Link 1 record, Link 2 record is also generated by the system to connect the title-record and author record having many-to-many relationships. The contents of Link 2 record are the data-base keys for the title and author records. The access to this record is always via title or author record.

7. Link 3 record :

This is also a link record connecting two keyword records. Unlike Link 1 and Link 2 records, Link 3 connect two records of the same type. The relationship between two keyword records is also of many-to-many type. This is because the keywords form a vertical structure in which a given keyword may have some keywords connected to it belonging to one level above and one level below it The keywords belonging to the level below are member keywords. The ones belonging to above level are called owner keywords. These two connections have been explicitely separated by Link 3 records. This is a typical case of bills - of - material type relationship in data-base terminology.

8. Publisher record :

This record consists of publisher-code (8 characters), publisher-name (maximum 200 characters) and country name. The records are sorted and stored in ascending order using publisher-name. However the individual publisher record has been stored and retrieved using calc key as publisher-code.

9. Sets :

The connections between various records are called sets in DBMS terminology. These sets play crucial role in fast retrieval of information. In fact identifying and establishing proper connections and then defining the

corresponding sets is an important aspect of data-base design. The sets that are defined in the system READFAST are described below :

i) Title-link 1 set

This set has title record as owner and link 1 records as mandatory members. This set implies that given an accession number, it is possible to find all link 1 records through this set. These link 1 records, in turn, will enable us to get all keywords associated to the accession number.

ii) Keyword-link 1 set

This set has keyword as owner and link 1 records as mandatory members. The set implies that given a keyword , one can get all its member link 1 records. Through these link 1 records one gets all the title records having a given keyword.

iii) Author-link 2 set

This set has author record as owner and link 2 record as mandatory members. This set helps us to retrieve all the title records connected to a given author record.

iv) Title-link 2 set

This set has title record as owner and link 2 records as mandatory members. This set helps us to retrieve all the author records associated to a given accession number.

v) Has-member set

This set has keyword record as owner and link 3 records as optional members. This set enables us to retrieve all the keyword records belonging to a level just below its node level. In other words a given broad keyword gets split up into number of narrow keywords.

vi) Is-member set

This set has keyword record as owner and link 3 records as optional members. This set enables us to retrieve all the keyword records belonging to a level just above its node level. In other words for a given narrow keyword, one gets all broad keywords of which the narrow keyword is a member.

In sets Has-member and Is-member, it may be noted that the membership is optional as it depends only on the hierarchy developed for a subject. In contrast, in other sets the membership is mandatory. For example a title-record will certainly have keyword and author records associated with it.

vii) Alpha-keyword set

This set has alpha record as owner and keyword record as optional members. This set connects alpha record and keyword records directly as it has one-to-many relationship. This set is helpful to get the list of keywords which begin with any given two letters.

viii) Alpha-author set

This set with alpha record as owner and author record as optional members connects them with one-to-many relationship directly. This set helps us to retrieve all author records for which surname begins with any given two letters.

ix) Sort-publ set

This set has system record as owner and publisher records as mandatory members. This set helps us to list all publisher names sorted alphabetically in ascending order.

10. Areas

READFAST system has been divided into four different areas.

i) Title-area : This area contains only title records.

ii) Keyword-area : This area contains keyword records along with link 1 and link 3 records.

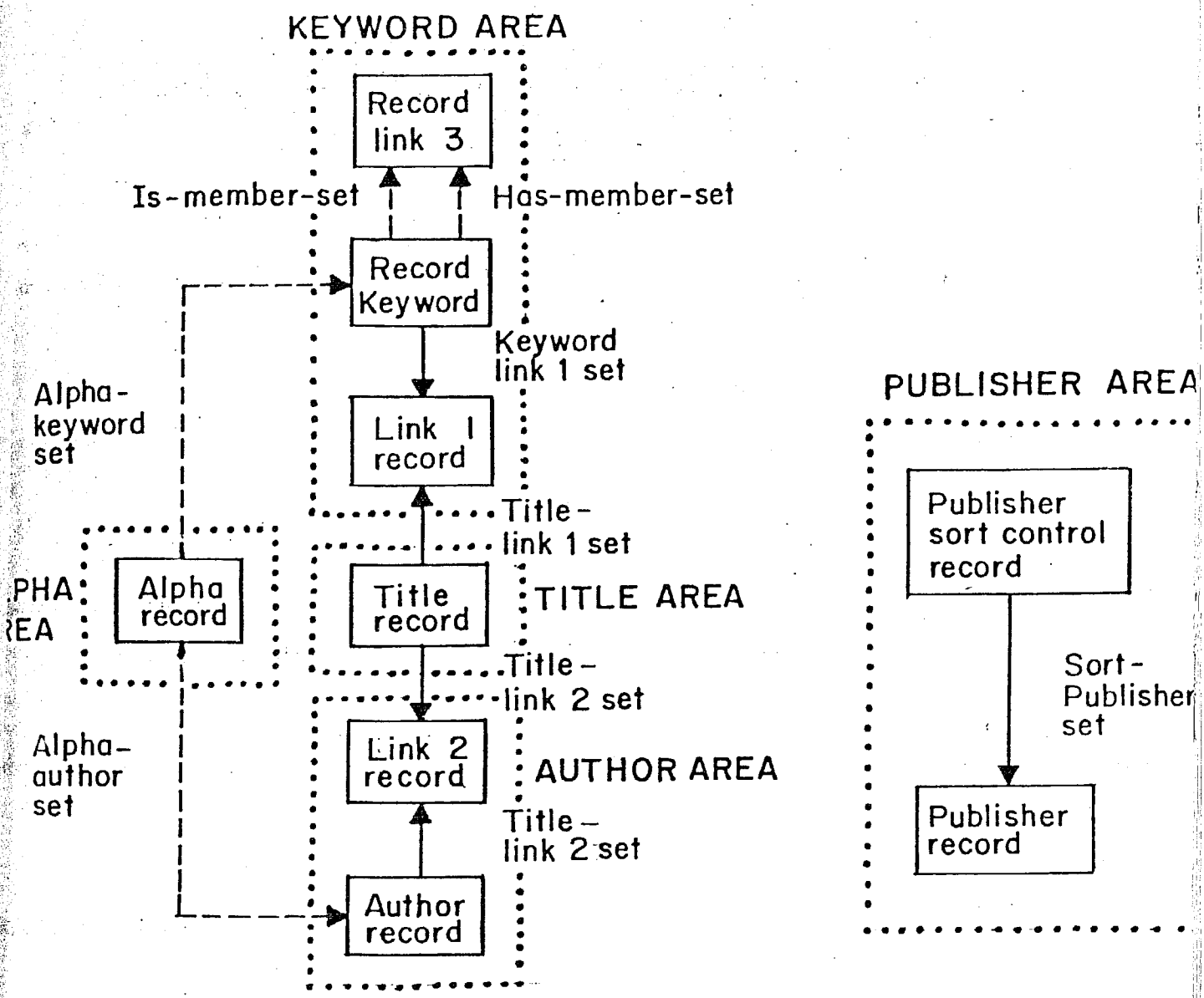
iii) Author-area : This contains author records and link 2 records.

iv) Publisher-area : It consists of system record and publisher records.

v) Alpha-area : This area consists of all alpha records

The figure gives the schema diagram showing various records, sets and areas used in READFAST system.

SCHEMA DIAGRAM FOR READFAST SYSTEM



OTTED BLOCKS REPRESENT THE DIFFERENT AREAS IN THE READFAS
ATABASE. THE SOLID ARROWS INDICATE THAT THE MEMBER RECORDS ARE
ANDATORY AND AUTOMATIC WHILE THE DASH ARROWS INDICATE THAT THE
EMBER RECORDS ARE OPTIONAL AND MANUAL.

VI. Structure of READFAST system

READFAST system consists of five independent modules.

Module No.1

This module consists of two programs. The first program is used for easy data-entry to create a raw disk file. The second one validates the data in the file and brings a print-out of the file in a format that facilitates manual checking. The file is then appropriately corrected if necessary.

Module No.2

This module loads the disk file created by module one in the data-base. After the file is loaded, it is restored on the tape for backup.

Module No.3

This module answers the user queries both simple and complex as described in sections 7 and 8.

Module No.4

This module is meant for updating the data-base as and when it is required. The details are described in section 9.

Module No.5

This module updates the circulation information about documents as described in section 10.

Major part of program coding has been done using FORTRAN, version 7 available on DEC-1091 machine. The version includes most of the facilities of ANSI FORTRAN 77. The system has been based on the data-base management package DBMS-10, versi 7 available under operating system TOPS-10. We use the DBMS package with FORTRAN interface both for data definition and data manipulation. There is, however, a slight difficulty to use the variables of type CHARACTER available in version 7 of FORTRAN. The DBMS package stores all character informatic in integer type variables which may be either scalar or array in nature. It is, therefore, necessary that one has to reconvert these integer variables again into variables of typ CHARACTER with appropriate length parameter. The system contains FORTRAN coding of approximately 5000 lines.

The system also includes one assembly routine written in DEC-1091 micro language. The routine gives the project and programmer number (PPN) of a person who has logged in as a string of 13 ASCII characters.

VII. Simple queries

Based on the schema developed in section 5, READFAST system offers various ways to retrieve the information from the data-base. The simple queries retrieve the information using only a single keyword or an author-name or an accession number at a time.Users can very well make use of these queries to enable them to browse through the data-base and make themselves familiar with various facilities existing in the

system. These queries are quite flexible and can be given in simple English sentences using a few reserved terms spelt correctly. Every query, however, should be terminated by a dot (.). The system can answer the following simple queries classified into five groupsfor the convenience of users.

Group I.

1. Get me keywords alphabetically.
2. Get me documents using keywords.

Group II.

3. Get me author-names alphabetically.
4. Get me documents using author-names.

Group III.

5. Get me documents using accession numbers.

Group IV.

6. Get me a list of broad keywords
7. Get me keywords linked to my keyword.

Group V.

8. Get me a list of keywords.
9. Get me a list of author-names.
10. Get me a list of all documents.
11. Get me a list of alpha records.

The verb 'Get' in above queries may be replaced by other similar verbs such as 'Fetch' or 'Obtain' or 'Find'.

1. General mode of operation

In this section we describe the general mode of operation for the simple queries listed above.

i) When you put a particular query, the system will ask for necessary input. If the input is not correct, it will ask for correct input. If the input is correct, the system will present to you the retrieved information in small logical segments. After each segment there will be a pause so that you can read the information on the screen carefully. In order to get the next segment, you simply press RETURN key. In this way you can get all the information. In case you want to quit the current query in the middle, you can type E and then press RETURN when there will be a pause. Thus for every query there are two exits One is the normal **exit** defined by the system. The other exit will be by typing E as described above.

ii) Once you come out of the current query, the system would like to know if you wish to have a hard copy for the same. By your affirmative response, the system will give you hard copy on the printer. Otherwise it will ask if you wish to put next query to the system. Thus you may continue the session as long as you want.

iii) You can terminate the session by typing STOP instead of putting the usual query.

2. Salient features of simple queries

We elaborate below the meaning and implementation of each query to point out its salient features.

The query numbers 1 and 3 will list out keywords and author-names beginning with any given two letters respectively.

The query numbers 8 and 9 will list all the keywords and author-names available in the data-base.

The query number 10 will list out all the documents giving the contents of title and author records associated with them.

The query number 5 will give all the details of the documents contained in title, author and keyword records for a given accession number.

The query number 2 retrieves all the title records corresponding to a given keyword. In response to this query the system will ask for your keyword. If the keyword given is not correct, the system will not entertain the query and will ask for correct keyword. After 3 trials, the system will automatically give the list of all the keywords available nearest to the last keyword given by the user.

The query number 4 retrieves all the documents for a given author-name. Here also the similar help is given by the system as in the case of query no.2, when

the author-name is spelt wrongly. In addition, if different authors have the same surname with different initials, the system lists out various initials when users feed only the surname. In case there is only one author with a given surname, the system does not require the initials.

The query number 7 helps the user to find out both the narrow and broad keywords linked to a given keyword. i.e. just the immediate top and narrow terms.

VIII Complex queries

Once the author-names and keywords are exactly known, one can handle complex queries which combine number of keywords, author-names and years of publications through AND and OR binary operations.

The general format for a complex query is given below.

Get me documents for (K=) (A=) (Y) (R=).

Where bracket K is a keyword bracket,
bracket A is an author-name bracket,
bracket Y is a publication year bracket,
and bracket R is range bracket.

Note that the query is always terminated by a dot(.). In query any type of bracket can appear more than once depending on the complexity of the query. The binary operator AND is indicated by dot (.) between the two brackets. However dot (.) is optional and hence need not be put explicitly. The binary operator OR is indicated by slash (/) between the two brackets.

The brackets K and A are either connected exclusively by AND operators or by OR operators. They cannot be, however, connected both by AND and OR operators in a given query.

Y bracket can contain connectors such as =, > and <

For example

(Y = 1970), (Y < 83), (Y > 76).

The first bracket will retrieve documents published in year 1970. The second one will retrieve those which are published before and in 1983. The third will get all the documents published after and in 1976.

R bracket contains range between two years connected by a connector word 'To'. For example

(R = 1980 To 84).

This bracket will get all documents published in the range of years 1980 to 1984 including both the years.

The year information can be given with four digits or with two digits. When it is given in two-digits, it is assumed to be the year in this century. For example

(Y = 75) is equivalent to (Y = 1975)

(R = 1970 to 72) is equivalent to (R = 1970 to 1972).

In any query the brackets K and A should always precede the brackets Y and R. Further the brackets K and A combined together are always connected to Y and R brackets by AND operator. This is because the year generally puts additional constraints similar to AND operator. Similarly Y and R

brackets are always connected among themselves by OR operators by the system even though the user may put AND operator explicitly between them. This is quite logical as the same document is generally not published in two different years or range of years. The mode of operations for the complex queries is very similar to that of simple queries described in the last section.

The following example will clarify the usage of complex queries further.

1. If one wants to know the work done on "solitons" in "quantum field theory" after 1980, one puts the qucry as follows.

   Get me documents for (K=solitons) (K=Quantum-field-theory) (Y>80).

2. If one wants to find a book on "Pascal programming" by N. Wirth, he may put the query as follows.

   Get me documents for (K=Pascal) (A=Wirth,N).

   In bracket A author-name is written with surname first followed by comma and then initials.

3. As stated earlier, the documents have been assigned even those keywords which reveal the level of treatment of the subject in the documents. Hence if one wants textbooks on astronomy, he can put a query as follows.

   Get me documents for (K=Astronomy) (K=Textbooks).

In practice the complex queries are quite powerful and handy in retrieving the inter-disciplinary documents as they eliminate noise and retrieve very precise pin-pointed information on micro-subjects i.e. subjects of small extension and therefore of great intension.

IX. Updates in READFAST system

Updating a data-base is generally the constant activity during the life of any data-base information system. In our case, however, the nature of data-base is such that except additions and deletions, the modifications in the data-base are not expected to be large. In fact a document with all its associated records when inserted correctly is likely to remain unchanged permanently in the data-base. So the main activity would be to add new documents and occasionally to delete a few which may be missing or lost. However keeping in mind some practical considerations, we have developed some additional procedures which may help the data-base administrator (DBA) to modify the data-base with ease.

The following procedures are currently available for updating the data-base.

1. Add a new document.
2. Delete a document.
3. Modify an accession number.
4. Modify an author-name
5. Modify a keyword
6. Modify a classification number
7. Modify issue-category of a document.

These updates can be effected easily using simple query-li mode by the DBA. We describe below the implementation of these update procedures in detail.

1. In adding a new document DBA is helped by the system at all crucial levels. In fact the system helps the DBA in data-entry, data validation and then loading the document in the data-base. A document with dupli accession number is generally not allowed to be loade in the data-base. However, a provision has been made so that it can still be loaded without modifying the existing record but inserting the additional informat in the record if any in the data-base at the same time. This provision can be made use of for effecting many small modifications that may be required in a partic record.

2. In deleting the accession number, care has been take to delete all its associated records. To be specifi the deletion of accession number involves deletion its title, author and keyword records. Further it a involves deleting number of link and alpha records related to above records.

3. In modifying the accession number the procedure also modifies the call number of which the accession numb is a part.

4. In modifying author and keyword records, the procedu also deletes if necessary the number of related alpha records. It also involves deleting the membership of previous records from alpha-keyword and alpha-author

XI. Impact of READFAST

Since we have used the DBMS package we can maintain data independence and also the integrity of the data, which would not have been possible in a non-DBMS environment.

The system developed is quite "user friendly". An essential and extremely useful feature of its design is that it not only provides descriptive assistance, but has also provided a structure that guides the user to the most useful answer. It is a system that has been fully self documented and hence the need for user training has been minimised.

Classification numbers get outdated from time to time. Re-classification is a time consuming and laborious process and even if two to three classification numbers are given to a book, it does not reveal all the detailed topics covere in the book and therefore classification numbers are not effective retrieval tools. We, therefore, need not reclassify books in future, since this system uses keywords which are powerful retrieval tools. May be the libraries of future could do away with the process of depth classification by arranging the material according to very broad subject groups, or any other easier arrangement.

Subject interest profiles of our scientists can be fed into the system in order to give Current Awareness Service which will list all the new material received about the

subject in which they are working. Very precise and selective information could also be disseminated to those users who want information for a specific proposal or project undertaken by them. This can be done very efficiently by this system due to detailed indexing which helps precision retrieval and eliminates noise.

At present we have various library subsystems developed such as Periodical Management System, Book Procurement System, Computer-aided-catalogs etc. We plan to integrate these so that it starts from the procurement stage to the processing and finally to the retrieval stage. The integrated system will ensure that data items are not duplicated for various sub-systems and this will also reduce the physical storage.

Acknowledgement

We would like to place on record our sincere thanks to our Director, Prof.S.P.Pandya; who has taken active interest and has always given us all the encouragement and support to complete this project.

It is a great pleasure to thank Mr.G.G.Dholakia and Mr.M.V.George for many useful discussions and system support that they have offered so willingly and generously. The entire Library Staff has been working on the reindexing, reclassification, recataloging, reinputing the entire data and all the follow-up work that is associated with this collosal task. This would not have been possible without their dedicated help and support.