IMPLEMENTATION OF ONLINE CATALOGS: DATABASE MANAGEMENT SYSTEMS VS. BIBLIOGRAPHIC RETRIEVAL SYSTEMS

LA MISE EN OEUVRE DES CATALOGUES ORDINOLINGUES: SYSTEMES DE GESTION DE BASES DE DONNES VS SYSTEMES DE REPERAGE BIBLIOGRAPHIQUE

Jean Tague, Ronald Davies, Anne Toombs
University of Western Ontario
SLIS
London, Ontario
N6A 5B9

ABSTRACT

For reasons of economy and convenience, many libraries would prefer to use existing software packages rather than developing their own implementations of online catalogs. This paper looks at the required and desirable features of an online catalog and compares the capabilities of bibliographic retrieval systems and database management systems to satisfy these requirements. In a pilot study at the School of Library and Information Science, University of Western Ontario, a sample of catalog users, including staff, faculty, and students, were interviewed concerning their needs in an online catalog. A test catalog was then implemented in two versions: one using GOTHIC, the in-house bibliographic retrieval system, and the other using DPL, a commercial database management system for the Decsystem 10 computer. The test catalog consisted of 10,000 titles in the SLIS collection. Benchmark tests consisted of 15 Boolean searches (5 search types, 3 replications). In terms of CPU time, GOTHIC proved to be faster, on the average, by a factor of 36:1.

RESUME

Pour des raisons d'économie et de commodité, plusieurs bibliothèques préféreraient utilisés des progiciels existants plutôt que de développer leur propre version de catalogue ordinolingue.

Les auteurs analysent les caractéristiques nécessaires et souhaitables d'un catalogue ordinolingue et comparent la capacité des systèmes de gestion de bases de données et des systèmes de repérage bibliographique pour satisfaire ces exigences. Dans un project-pilote mené à la School of Library and Information Science (SLIS) de l'University of Western Ontario, Un échantillon d'utilisateurs du catalogue, incluant le personnel, les professeurs et les étudiants, furent interviewés pour connaître leurs besoins dans un catalogue ordinolingue.

Un catalogue expéri mental fut ensuite conçu en deux versions: l'une utilisant GOTHIC, le système interne de repérage bibliographique et l'autre utilisant DPL, un système commercial de gestion de bases de données pour l'ordinateur Decsystem 10. Le catalogue expérimental consistait en 10,000 titres de la collection de SLIS. Les tests repères consistèrent en 15 recherches booléennes (5 recherches-types, 3 essais). En terme de temps CPU, GOTHIC s'est aviéré plus rapide, en moyenne dans une proportion de 36:1.

The 1970's saw the emergence of the microcatalog. As libraries move into the 1980's however, the trend is towards online catalogs. Several large research libraries are already investigating, testing, and planning for these, as reported, for example, by Veneziano (1980). For some libraries, online catalogs will evolve from commercial minicomputer-based circulation systems. Some vendors of such systems, such as GEAC, CLSI, and UTLAS, are adapting them to provide at least minimal online catalog capabilities. Other libraries, particularly those which form part of a larger institution, look to multi-service computer centers within their own environment to provide the facilities needed for the implementation of an online catalog.

For reasons of economy and convenience, many libraries would prefer to use existing software packages on such systems, rather than developing their own programs. Software development requires an extensive outlay for programmers and CPU time and this is followed by equally extensive time and money investments in the testing, debugging, and maintenance of the software.

Two existing types of software packages -- bibliographic (or information) retrieval systems and database management systems, are candidates for implementing online systems. Both are examples of information systems, which Salton (1975) defines as computerized collections of records, stored and processed to provide information to user populations.

The Codasyl Committee (1976), describing the common features of database management systems, defined data as "the representation of facts or the recording of events in a formalized manner for the express purpose of communication, interrogation or processing by either humans or machines." A database management system is then a software package which provides a set of programs processing the collection of data or database through a common user interface. Typically, the set of common functions performed by the programs include definition, creation, interrogation, and update. Thus, in short, a database management system accesses a computerized store of formated information for various purposes. By formated is meant information which can be specified by a fixed number of clearly defined fields. Initial applications of database management systems lay in the areas of personnel and inventory systems in business, where the items of concern--people or parts --could be described for purposes of the business applications by employee number, name, department, salary, deductions, etc., in the first case, and by part number quantity on hand, supplier, etc., in the second.

A bibliographic or information retrieval system, on the other hand, accesses a computerized store of textual information. Textual information, in general, is at least partially non-

formated and in natural language form. Each item cannot be easily described by a constant number of fixed-length fields. Commercial bibliographic systems such as DIALOG, ORBIT, BRS, CAN-OLE, Q/L INFO GLOBE, provide access to such nonformatted information as titles and abstracts of journal articles, full text of newspaper articles, and full text of statutes or case law reports. Bibliographic retrieval software is designed to enable the user to search the text collection to find references and passages of interest. Although the term database is sometimes applied to the stored collection accessed by a bibliographic retrieval system, it is more correct to call it a textbase, since it contains free text as well as formated information.

FEATURE ANALYSIS

A number of writers, notably Olle, (1971) Huffenberger and Wigington, (1979) and Town and Powell, (1980) have compared the features of database management systems and bibliographic retrieval systems. Olle distinguishes between generalized systems, which can be used for a wide range of applications, and tailored systems, which are built for a particular application. In a tailored system, the meaning of each field is built into the system. Most present-day database management systems and bibliographic retrieval systems can be considered as generalized systems, since both can handle a wide variety of applications. Both can also be considered tailored, however, since database management systems are tailored for applications involving easily formated data (a point Olle seems to have missed) and bibliographic retrieval systems are tailored to applications involving textual data. As generalized systems, both provide a user interface language which makes it possible for a user to define the logical structure of the data or text base and manipulate it logically in a high-level language without being concerned with details of machine-level storage and programming. In both, processing commands are independent of the structure of the data or text upon which they are acting.

Database management systems permit more complex hierarchical record structures than do bibliographic retrieval systems. Although some permit multiple record types with a file, for each type, the fields are unique, fixed in number and in length. In a bibliographic retrieval system, records are not usually hierarchically structured, but they can be variable format, with variable length fields, repeating fields, and optionally occurring fields. For example, in a personnel record, the fields name, number, salary, department will always be present and will contain data in a standard length and form--for example, the name will be allotted exactly 20 characters. In a bibliographic record describing a journal article, there may be zero, one, or more than one personal author. The title and abstract lengths will be so variable that it would be extremely inefficient to describe them in terms of a fixed length field.

Most bibliographic retrieval systems are interrogated online because of the need of users to search interactively or to browse through the files. The reason for this need is that users of text files usually want information on a particular topic which may or may not be exactly expressible in terms of keywords or terms in the text. The output from a search will not be an answer to a query, but merely a description or references which may or may not be relevant to the topic. By an iterative process of refining and expanding the search statement, the user (or an intermediary) finally arrives at a more or less satisfactory output. Users of data files, on the other hand, approach with a specific request in terms of known field values, and are interested only in records which match these requirements exactly. Since browsing is not important, database management systems may be online or batch, although the trend is to the former, because of its convenience.

The principal function offered online by the bibliographic retrieval system interface is interrogation, which normally means entering criteria for selection of records and formating of output. Typically, searching in bibliographic retrieval systems involves complex Boolean combinations of individual terms or keywords or phrases (adjacent terms) within a field. Although database management systems permit interrogation, criteria must usually be specified in terms of complete fields, rather than terms within the field. Greater emphasis is placed on report generation than on retrieval.

Updating and maintenance are functions more strongly supported in database management systems than in bibliographic retrieval systems. Usually it is assumed, with the latter, that these functions will be performed by specialists at a centralized center, where interrogation is performed by non-specialist users at remote access points. Data integrity, security, and privacy controls are, again, more important for database management systems than for bibliographic retrieval systems.

Any system which provides a generalized approach will pay for the extra layer of software in terms of increased computer time and storage costs. The trade-off, in deciding whether or not to use such a system is between machine time and people time. The interface language, at least in theory, provides a fast, simple access to the system for someone with little programming expertise.

Online library catalogs have some of the characteristics of database systems and some of the characteristics of textbase systems. Some of the information in a record is easily formated--call number, imprint, collation. Other information is variable length-- title--or optionally occurring--series title, analytics--or repeating-- subjects, authors, subjects. Searches

may be for specific information--i.e. known item searches--or for references on a topic--i.e. subject searches. A library director wishing to implement an online catalog using existing software packages may be puzzled to know whether to opt for a database management system or a bibliographic retrieval system.

SYSTEMS COMPARISON

Few empirical comparisons of database management and bibliographic retrieval systems have appeared in the library and information science literature. Town and Powell (1980) have looked at the application of the commercial database management system ADABAS to a scientific data bank containing information on environmental chemicals (ECDIN). The bank had previously been accessed using a retrieval system called SIMAS. The system was converted to ADABAS because it was felt that data management was a more important criteria for the selection of a system than the retrieval function. Although the ADABAS retrieval function was inadequate for a scientific data bank, it was felt it could be supplemented with relative ease.

School of Library and Information Science at the University of Western Ontario is considering the conversion of its library catalog to online form from its present microfiche form. A user survey was carried out to determine the necessary and desirable features in such a catalog. Following the survey, in order to provide some guidance about the relative capabilities and efficiency of database management systems and bibliographic retrieval systems for this application on the University's DecSystem 10 computer, a test catalog of 10,000 titles (approximately one-quarter of the collection) was put on disk in two versions: one using GOTHIC, the in-house bibliographic retrieval system, and the other using DPL, a commercial database management system for the DecSystem 10. Benchmark tests were then run on the two systems. The remainder of the paper describes this pilot study. Before giving the results of the pilot study however, we shall provide a brief description of the two systems.

DPL (Data Processing Language) is a database management system developed for the DecSystem 10 timesharing computer by National Information Systems, Inc., of Cupertino, California. Written in macro assembler language, it has the usual database management features including a data definition language and data manipulation commands covering all common data processing functions--managing, querying, updating, reporting, and maintenance. In addition, with its "processes" and "procedures", it has many of the capabilities of a programming language. It permits four different types of indexes to information in a field--pointer, hash, key, and isam and allows access, in addition to the file currently opened, to information in 8 tables.. Files in DPL may contain records of fixed, counted, or

variable length. In searching, Boolean combinations of field values may be specified and searching may be done on parts of a field. However, searches on parts of a field (keyword searches) cannot utilize key or other types of index, and this will make searches of large files extremely slow, since it must be done in a sequential fashion.

GOTHIC (Gulatzan's Online Thaumaturge and Helpmate in Information Control), also written in the macro assembler language, is modeled on commercial systems such as ORBIT and DIALOG. It accepts records in the form of variable length fields with three-letter tags defined by the user. In one record, tags may be repeated as often as needed. Four indexing options are available for each field: "normal" indexing, in which every word is treated as a keyword and indexed; indexing with a stop list: indexing with a special key composed of the first letter of each of the first twelve words in a field; and no indexing. GOTHIC accepts input from the terminal or a file and automatically builds three levels of keyword indexes which are used in retrieval of items from the master file. The indexes can be built automatically as data is added to the master file or later in batch mode. However, users have encountered serious problems with some of the updating functions. The lack of program documentation and the unstructured nature of the programming have made the GOTHIC system difficult to maintain. Its searching capabilities, however, are sophisticated. Searches may specify any Boolean combination of keywords from any field and set manipulation and truncated search terms are possible.

The user survey involved interviewing 24 faculty, staff, Ph.D. students and M.L.S. students at SLIS who were familiar with online systems. To some extent, the sample was self-selected, but it was felt that to interview those with no experience with online systems would provide little useful information. Interviewees were asked what features of an online catalog they would most likely use, what fields would be searched and printed out most often, how often the catalog should be updated to suit their needs, and what kinds of formating capability they would like. Some of the questions were open-ended and some were answered on a numeric scale representing frequency of use: 5--always, 4--frequently, 3--occasionally, 2--seldom, 1--never. The results of the scale questions were tabulated using the MINITAB Statistical Package on the DecSystem 10.

The decision as to which features should be included in the test catalog was based on the median response for each scaled question. If the median was above a specified cut-off, the feature corresponding to that question was included. The results showed that the most-used features, with medians of 5 or 4, would be Boolean logic, set manipulation (i.e., the ability to create a set containing results of a previous search and combine it with other search specifications or to combine several sets, each

containing the results of previous searches), browsing a keyword or other index, abbreviations of commands, adjacency (phrase searching), unlimited right truncation, searching parts of fields (i.e., keyword searching), and relational operators. If a similar median cutoff had been used in deciding the fields to be indexed, the only searchable fields would have been author and title. Therefore, a median of 3 was used with the result that searchable fields included call number, added author, added title, previous title, succeeding title, and the location code. A similar set was obtained for the fields to be output, again using a median cut-off of 3. A default display format was provided which includes the author, title, call number, and location. The median answer to the question "How inconvenient would it be to have to specify a field to be searched as well as a term?" was 2, i.e., slightly inconvenient.

The survey also attempted to determine the optimal time between updates of the main catalog. To the question of how much it would interfere with their work or studies if the online catalog were updated once every three months, the median answer was "somewhat"; for the rest of the intervals-- one month down to three days--the median answer was "not at all". The second question about updating tried to determine at what point people feel that they wouldn't need to search a separate file of new acquisitions as well as the main catalog; in other words, what would be the ideal interval between updates of the main catalog? The median response suggests that users would search the separate file frequently if the main catalog were updated once every month, seldom if it were updated once every two weeks, and any shorter intervals between updates would not necessitate searching the separate file at all.

In the open-ended questions, responses were mixed. There was fairly equal distribution of answers to the question about the order references should be printed at the terminal; by date with most recent first was the most popular. A majority also preferred having field tags appearing in the printout, there should be one field to a line in the print-out, and a space between each record. A majority were also in favor of having search terms highlighted.

Users were asked their reactions to GOTHIC and DPL, if they had had experience with these systems. The main complaint against GOTHIC was its tendency to break down in the manipulation of large files or during any update procedure. Other complaints included: 1) the necessity of using a system SORT and MERGE when updating or creating a database; 2) some inhospitality in the query language, in that misplaced blanks will nullify commands; 3) restricted field lengths; 4) fields must be specified for output.

GOTHIC was praised highly for its flexible, speedy search capabilities. It was found easy to learn and to use, while, at the same time, being capable of performing complicated functions. Users liked the fact that they could search either by keyword or by specifying a field, that they could manipulate sets, use Boolean logic and truncation, and have control over the format of output. Its similarity to commercial retrieval systems such as DIALOG or ORBIT make it an excellent learning tool for students.

Those who were familiar with DPL pointed out that it was not created for manipulating bibliographic data effectively, and that therefore it is inferior in this respect to GOTHIC, which was specifically designed for these applications. DPL was accused of being slow and expensive for small searches. Some found that the vast amount of documentation that is available for DPL, while a necessary adjunct, tends to be confusing for the novice user. Such limitations as not being able to search all fields at once, awkward use of Boolean logic, and not being able to update tables were disliked by those interviewed.

The fact that DPL has a variety of capabilities and that one can program it to adapt to specific needs were popular features of DPL. Its ability to handle large data files was a strong point; users found it easier to create, merge, and sort databases and to search on fields in DPL than in GOTHIC. Unlike GOTHIC, DPL is constantly being revised and improved.

SYSTEMS TESTING

Most bibliographic retrieval systems are based on an inverted file structure which permits rapid access to keywords within fields. As noted above, this type of indexing is not possible at present directly with DPL. For some time, DPL users at the University of Western Ontario had speculated that it might be practical to write a "front-end" program for the DPL system which would provide the capability for fast keyword searches based on an inverted file structure. One purpose of the pilot study was to demonstrate the feasibility and practicality of such a front-end program.

Such a program would require a number of different modules if it were to truly emulate a bibliographic retrieval system. Figure 1 shows the over-all structure. It was well beyond the scope of the pilot study, which was restricted to a three-month period, to develop such a system in its entirety. However, it was decided to attempt to implement simple search algorithms in a rudimentary front-end program. By then running a series of benchmark tests with a library catalog database, it hoped to get some idea of the efficiency of such an approach.

The choice of the SEARCH and PRINT modules was deliberate. The most common operations on an online catalog are searching and

updating, but it is in the searching area that database management systems are weakest as compared to bibliographic retrieval systems. If DPL could be adapted to perform effective and efficient searches it would certainly deserve further consideration; if DPL was not capable of efficient information retrieval, it could in effect be ruled out as a system for maintaining an online SLIS library catalog.

Evaluating DPL for the management of a library catalog was difficult because of the sheer number of options and alternate approaches possible. There are few guidelines given in the documentation to assist the applications programmer in making the right decision in terms of a specific applications. Testing each of the choices available would be an extremely time-consuming task. We relied on what information was available (e.g., indexes should not be formed for data sets of less than 2,000 records); on algorithms discussed in the literature (e.g. using post-fixed order to evaluate search expressions); and on our own judgment and the results of the survey. It was decided to store the catalog in binary variable format since this permitted the packing of records which are in fact variable length, i.e., with embedded blanks. The inverted keyword file, on the other hand, was stored in binary count format, since it is frequently searched and has blank fields only at the end of records. A key index was used for the inverted keyword file which was created from the machine-readable catalog in the initial processing.

In the SEARCH program, a DPL data set (file) was used to contain the document reference numbers relevant to each search term. These datasets were then compared according to the Boolean logic of the search string using DPL data manipulation commands. Incorporation of data manipulation commands in the logic forced the use of DPL procedures rather than DPL processes, and this choice, because procedures are interpreted rather than compiled, may have resulted in a lower efficiency.

The two systems used in the benchmark tests are not completely alike in their capabilities. GOTHIC contains other modules besides SEARCH and PRINT. As well, the DPL SEARCH program lacks several features found in GOTHIC, notably set manipulation, field specification, and unlimited right truncation. It appears that it would be possible to implement set manipulation without too much difficulty, particularly with the addition of the FIND command to the DPL repertoire. Field specification and especially unlimited right truncation, on the other hand, present many difficulties. To implement either or both would have lowered the over-all efficiency of the SEARCH program considerably.

The original intention of the pilot study had been to run benchmark tests against the full catalog of 40,000 records. However, it was not possible to obtain enough disk space to

accommodate such a dataset and its indexes. Instead, a small title catalog of 10,000 records was created from the SLIS library catalog, constituting a minimal catalog of approximately one-quarter of the collection. The actual test consisted of five groups of three search strings each. Search strings ranged from simple single term expressions to complex combinations of ORs and ANDs, with embedding. (See the Appendix for the actual strings used.) For each group, the system was called from monitor level, three search strategies of a type run, and then an exit made to monitor level and statistics recorded. Finally, two of the search strings were searched independently once, and were searched again with retrieved catalog entries printed out at the terminal to test printing efficiency.

The benchmark test results suggest that while DPL and the SEARCH front-end program required less disk storage than the comparable GOTHIC database, cost for building the database, and especially for searching, were considerably higher. The mean search time for the five groups of three search strategies each was 36 times higher with the DPL SEARCH program than with GOTHIC. The full results of the benchmark tests are given in Table 1. While improvement in the SEARCH program might reduce this difference, the discrepancy is great enough to make contradiction of the general conclusion unlikely.

The practicality of a front-end bibliographic retrieval program to the DPL database management system is seriously called into question by the benchmark test results. Development of a complete online catalog inquiry system that would meet the needs of the SLIS community would involve a considerable programming effort. As well, the tests suggest that such a front-end program would retrieve information extremely inefficiently when compared to a program, like GOTHIC, specifically designed for bibliographic text bases.

Since the tests, DPL has implemented the FIND command, which improves capabilities for keyword retrieval to some extent. However, before DPL and other state-of-the-art database management systems can really be considered viable routes for online catalogs, they will need to implement in full the features discussed in this report: full within-field keyword searching using a KEY-type index to individual terms, possibly with a specified stop list; search queries which can incorporate the results of previous searches (set searches); and unlimited right truncation.

Online catalogs and other interactive bibliographic applications are becoming increasingly important in business, government, and educational institutions. It is recommended that software houses and computer manufacturers in the near future should pay great attention to the development of efficient bibliographic retrieval systems or integrated database management

- bibliographic retrieval systems. The over-all design of such a system was presented by Schek, (1980) but unfortunately there is no sign that it will be developed commercially. As well, it is the responsibility of the information science community, for which this body is the spokesman in Canada, to make its needs known to the manufacturers and software producers. This study is intended to be a step in that direction.

BIBLIOGRAPHY

Codasyl Systems Committee. Selection and Acquisition of Data Base Management Systems: Report. N.Y., Association for Computing Machinery, 1976, 252 p.

Huffenburger, M.A. and Wigington, R.L., "Database management systems," in Annual Review of Information Science and Technology 14, vol. 14, (1979), pp. 153-190.

Olle, T.W., "A Comparison between Generalized Data Base Management Systems and Interactive Bibliographic Systems," in Interactive Bibliographic Search: the User/Computer Interface, Montvale, N.J., AFIPS Press, 1971.

Salton, Gerard, Dynamic Information and Library Processing, Englewood Clifts, Prentice-Hall Pres, 1975, 523 p.

Schek, H.J., "Methods for the Administration of Textual Data in Data Base Systems." Internation Conference on Information Storage and Retrieval, 3rd Cambridge England, 1980. (to be published in 1981 by Butterworths under the title Information Retrieval Research). Town, W.G. and Powell, J., "Recent Experience of the Application of a Commercial Data Base Management System (ADABAS) to a Scientific Data Bank (ECDIN)," in Information Processing and Management, vol 16 (1980), pp. 91-108.

Veneziano, Velma, "Library Automation: Data for Processing and Processing for Data." in Annual Review of Information Science and Technology, vol 15 (1980), pp. 109-46.

APPENDIX

Search Strategies Used in Benchmark Tests

1. Single Term
   a) ACQUISITION
   b) MOZART
   c) ADMINISTRATION

2. Two Terms (&)
   a) LIBRARIANSHIP & WOMEN
   b) LIB & INFO
   c) GOVT & PUBN

3. Two Terms (+)
   a) ARCHIVES + ARCHIVAL
   b) JOURNAL + PER
   c) POETRY + POETICS

4. Three Terms (&)
   a) CLASSIFICATION & DEWEY & DECIMAL
   b) IRISH & FAIRY & STORIES
   c) LIB & AUTOMATED & SYS

5. Combinations
   a) REF & (QUERY + (QUESTION & NEGOTIATION))
   b) (BUDGETS + BUDGET + BUDGETING) & (LIBRARIES + ADMINISTRATION)
   c) (SPECIAL & COLLECTION) + (RARE & BOOK)

TABLE 1

SUMMARY OF BENCHMARK TEST RESULTS

Test	DPL Front-end	GOTHIC	DPL/ GOTHIC
Storage Space (in blocks)	2130	2411	.883
Search Time (in CPU sec.)			
Group 1	9.18	0.25	36.7
Group 2	16.75	0.76	22.0
Group 3	13.85	0.26	53.3
Group 4	21.41	0.66	32.4
Group 5*	29.50*	0.51	57.8*
Total	90.69	2.44	37.2
Search & Print (2(b) in CPU sec)	8.61	2.36	3.7
Search alone	7.78	0.66	11.8
Print alone	.83	1.70	0.49
Search & Print (5(c) in CPU sec)	9.76	0.45	21.69
Search alone	9.58	0.35	27.37
Print alone	0.18	0.10	1.8

*Program used for DPL SEARCH was not exactly the same as program used for other groups. The SEARCH program was revised slightly to avoid a bug.

GENERAL SYSTEM STRUCTURE

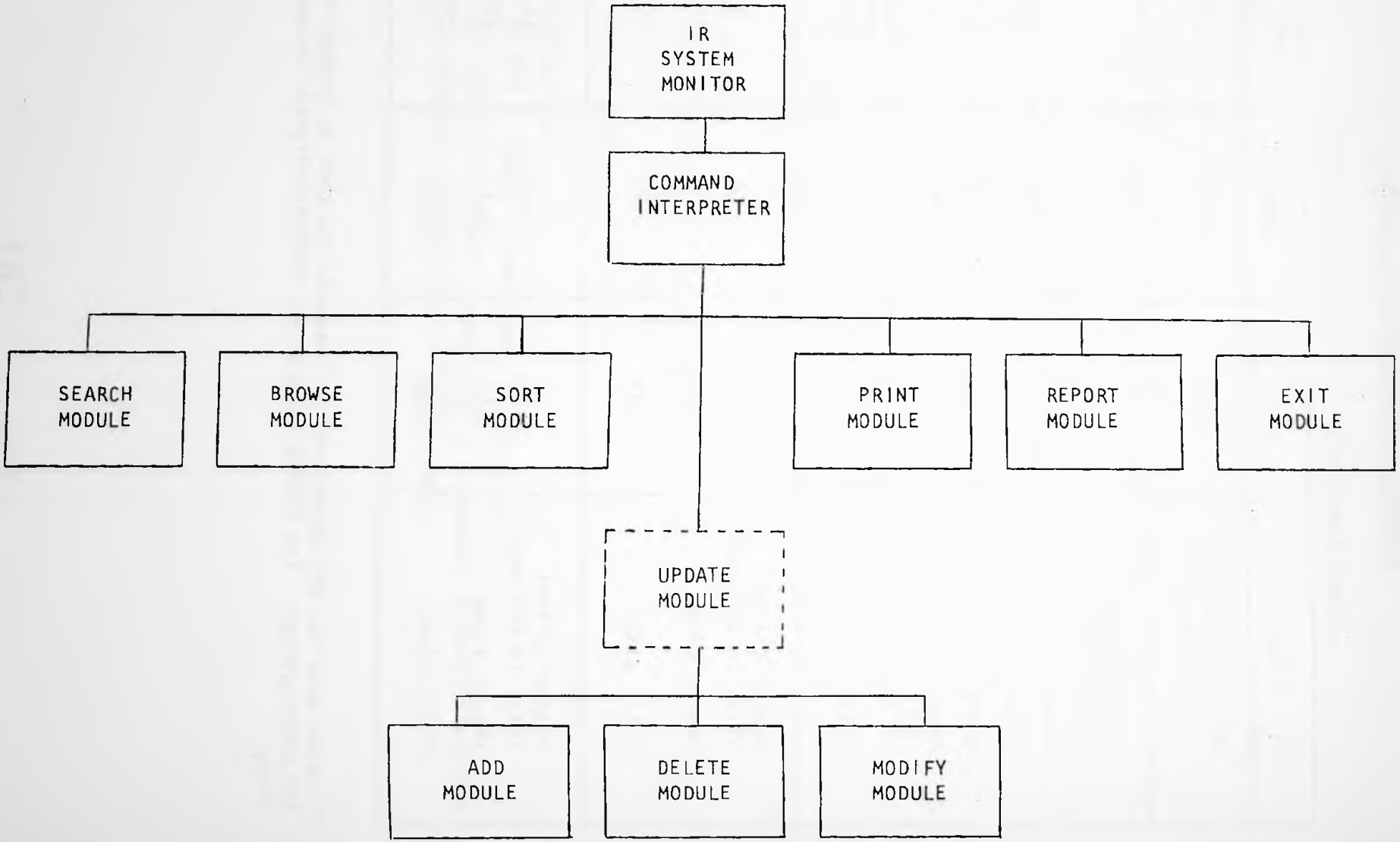

FIGURE 1