INFORMATION SYSTEMS

Dr. C. J. Bell

IBM Scientific Centre,
Neville Road,
Peterlee,
County Durham.

Abstract:

The nature of information systems is introduced and their historical development traced to the systems of today. The importance of information systems in the future application of computers is stressed and the major difficulties lying along the road to progress are discussed. Not the least of these is an adequate theoretical characterization of information systems. The treatment is introductory and a reading list is appended.

1. Introduction

1.1 Characterization

An information systems is a model of a complex functioning environment providing a mechanism for transmitting the right information to the right place at the right time in the right form. Effective management, control and planning of complex systems behaviour is dependent on a comprehensive and well-organized information system.

The problems of constructing an information system are determined by the high volume of information within the system, the activity in the system, the complex transformations required and the speed of transmission to and from remote locations.

Although it is not necessary to have a computerized information system, especially in low volume situations, it is hard to envisage really effective management of complex systems not being able to benefit immeasurably from computerization. This presentation is concerned with the software development orientated towards the implementation of information systems.

1.2 Example - the IBM RESPOND system

The management of the operations in the business of the IBM Company provides a web of such management information problems. In the co-ordination of customer orders, for example, we have a situation wherein orders for complex computer systems are generated continually in 105 countries of the world. These are concentrated in 19 major marketing locations and the order is split up so that separate component orders are placed at manufacturing plants in the U.S.A. and elsewhere. The speedy and efficient resolution of these orders to optimize delivery dates and production schedules is crucial to the IBM business.

Within the IBM World Trade Corporation, all such orders are entered through the computerized RESPOND systems (standing for Retrieval Entry Storage and Processing of On-line Network Data). Large centralized data-banks are maintained at the RESPOND Centre in Havant, Hampshire. These are connected by an international telecommunications network to the other operating points where orders are generated and placed.

In addition to the above manufacturing order entry system, other aspects of IBM business are driven by an information system such as DP marketing, customer engineering, office products, etc. The overall

result is significant improvement in service through better management and control of the operation. However, an even greater pay-off potential is offered by the integrated data-banks providing accurate and timely information for forecasting demand and planning more efficiently for the future.

1.3 Types of information system

The information base is the aggregation of information maintained within the system. It may take a prescribed and well defined form resulting in some regularity in the structure of the information. It is then commonly referred to as formatted data and, in fact, is an organized set of data depicting the information which it represents. The other distinguishable form - referred to as unformatted data - occurs as text, being a collection of sentences or documents. Hybrid information bases are combinations of these two forms. The query language can also be categorized into three types. The first is formal having well defined set of sentence forms together with prescribed resultant action for each. The second is a natural language form where some analysis of a query must be performed to determine the meaning — the action to be performed by the system to obtain answers. The third type is a deductive query language. In this case a query response action can become complex since answers are sought not only by extracting appropriate data from the base but also by developing consequences of existing data.

The prescription of the query language assumes some form or other of the information base. Three categories of systems can be distinguished.

The first of these is a document retrieval system. Ideally, a base of natural language documents subjected to interrogation by a natural language query is desirable. Accepting the difficulties of machine resolution of a natural query and matching it to a natural language document, the normal course of action is to design a system to retrieve a subset of the documents wherein there is a high likelihood that the answer will be found. Final resolution must be performed by human scrutiny. Some systems maintain a natural language form for the base — usually document abstracts — but more commonly a document is represented by a set of index terms. The query language is usually some logical combination of index or key terms and the matching algorithm can be quite complex. The ability of such a system to retrieve 'all and only' the

relevant documents is limited, but such systems are finding increasing use in industry and commerce as the utility of the limited function becomes justified by reducing costs. The second type of system is termed a fact retrieval system. Here the intent is to achieve a high degree of accuracy in providing exact answers to specific questions. Query languages usually incorporate some deductive capability and the aim is to provide a controlled natural language subset as the medium for both information base and query language. At this time, fact retrieval is the subject of experimentation and research, primarily in the universities, especially in the U.S.A. There are many workers who have expressed pessimism as to the ultimate feasibility of ever developing a high quality fact retrieval system on a large enough scale to have any significant impact in practice. The third type of system is a data retrieval system providing effective means of posing formal queries to a formatted information base. This type of system has been in wide use for a decade or so and offers the major area for increased popularity in the future. The remainder of this paper will be concerned with information systems of this type only.

We shall begin by introducing the terminology and showing how a conventional system is built up. The influence of hardware developments on systems design will be mentioned and future prospects and problems discussed.

2. Conventional systems

2.1 File systems

A conventional system is built around the notion of a file which is regarded as a collection of records each of a similar morphological construction. The information base is then the set of all such files.

In order to provide an organization for more efficient response the set of records in a file is usually ordered. The components of an information system are shown in figure 1 and provide the essential means for defining a file, creating it, maintaining it and querying it.

* Data Description Language
* File Create Facility
* Update Facility
* Index or Sort Facility
* Query Language

Figure 1: INFORMATION SYSTEM COMPONENTS

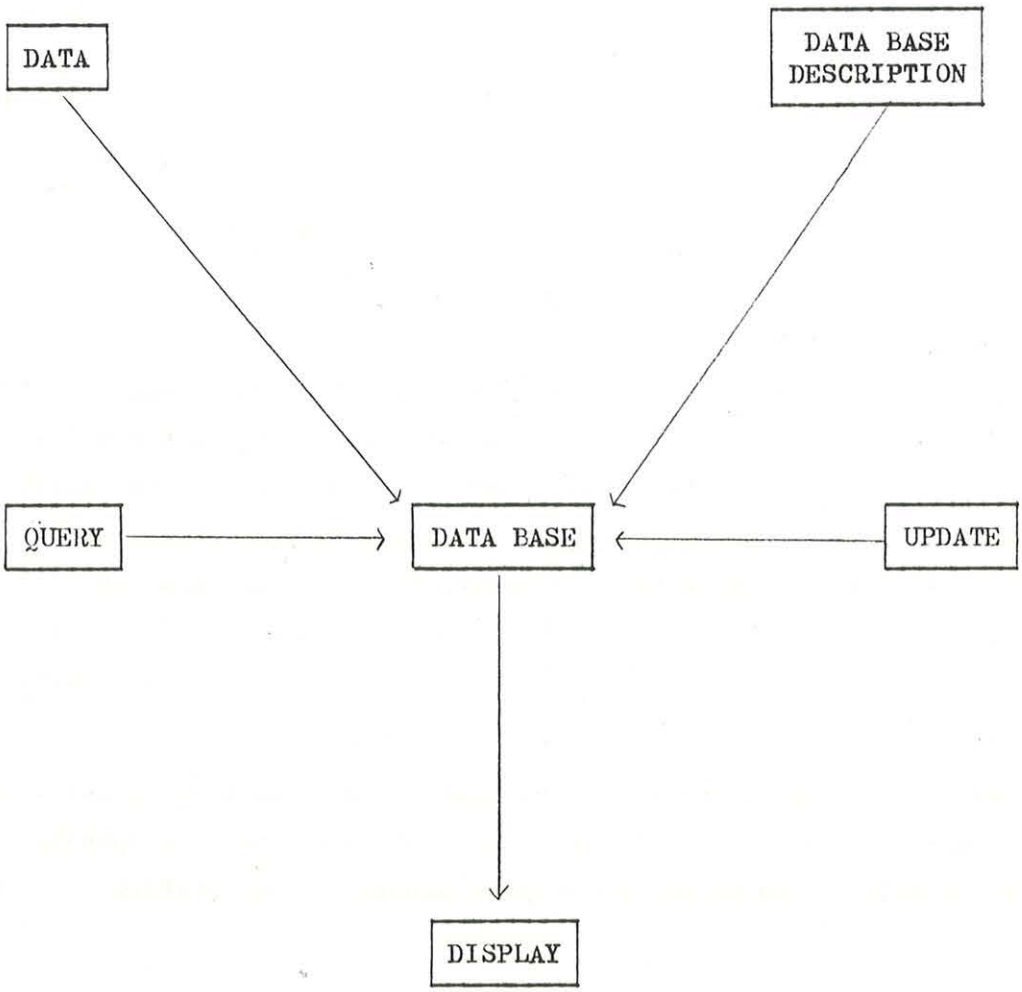

Figure 2: INFORMATION SYSTEM STRUCTURE

In figure 2 the relationship of these components is displayed. The information contained in and form of a typical record of a file is given by a set of statements in the data base description (DBD) presented to the system in the data description language (DBL). Each record is said to consist of a collection of fields designated in the DBS by a field name, being a valid name of the DBL. Each field specifies that each record will contain an elementary data item or field value of a certain type — the field type — e.g. a real or integer value or a sequence of characters of fixed or variable length.

One field is usually earmarked as the key field and records within a file are sequenced in order of occurring field values of that field. In a file of employee data, for example, the field with field name EMPLOYEE NUMBER might be designated as the key field and the records sequenced in increasing order of such a number. In this case, there will be one record for each employee. 'Employee' is thus a preferred object and it is called an entity. In addition to constructing a record from fields, the concept of a repeating group or segment is encountered. This is exactly equivalent to a sub-file dependent on the record of which it is a part. It will consist of a sequence (possibly ordered, and possibly indefinite in number) of sub-records which must also be defined. In the employee record, for example, we may wish to incorporate JOB HISTORY as a repeating group, being a set of records describing each JOE held by the employee.

In turn, a repeating group record may also contain one or more repeating groups dependent on it. We thus encounter a hierarchically structured data file of potentially unlimited extent. In practice both the number of repeating groups at each level and the number of levels is usually severely restricted.

A third type of unit from which a record might be notionally constructed, is called a virtual field. In this case, no actual field value is actually stored. The value is computed from other field values — real or virtual — by a well-defined procedure prescribed in the DBD. If the employee record contained a repeating group termed SALARY HISTORY giving the salary at each change through time, then AVERAGE SALARY could be defined as a virtual field and computed from the elements of this repeating group when required.

1.	Employee Name	(Name)
2.	Employee Number	(Number)
3.	Birthdate	(Date)
4.	National Insurance Number	(Number)
5.	Marital Status	(Name)
6.	Organizational Unit	(Name)
7.	Current Salary	(Number)
8.	Number of Children	(Number)
9.	Job History	(RG)
10.	Name of Technical Speciality	(Name in 9)
11.	Years in Speciality	(Number in 9)
12.	Technical Speciality History	(RG in 9)
13.	Job Title	(Name in 12)
14.	Date Title Received	(Date in 13)
15.	Salary History	(RG in 12)
16.	Salary	(Number in 15)
17.	Date of Last Increase	(Date in 15)
18.	Total Years of Education	(Number)
19.	Educational History	(RG)
20.	School or College	(Name in 19)
21.	Degree	(Name in 19)
22.	Year of Degree	(Number in 19)
23.	Course	(RG in 19)
24.	Title of Course	(Name in 23)
25.	Grade, Position	(Name in 23)
26.	Occupational Proficiency Test	(RG)
27.	Date of Test	(Date in 26)
28.	OPT Score	(Number in 26)
29.	Occupational Aptitude Test	(RG)
30.	Date of Test	(Date in 29)
31.	APT Score	(Number in 29)

Figure 3: PERSONNEL FILE

An example of a hierarchically structured file is given in the DBD displayed in figure 3 and the hierarchical nature of the structure is given in figure 4.

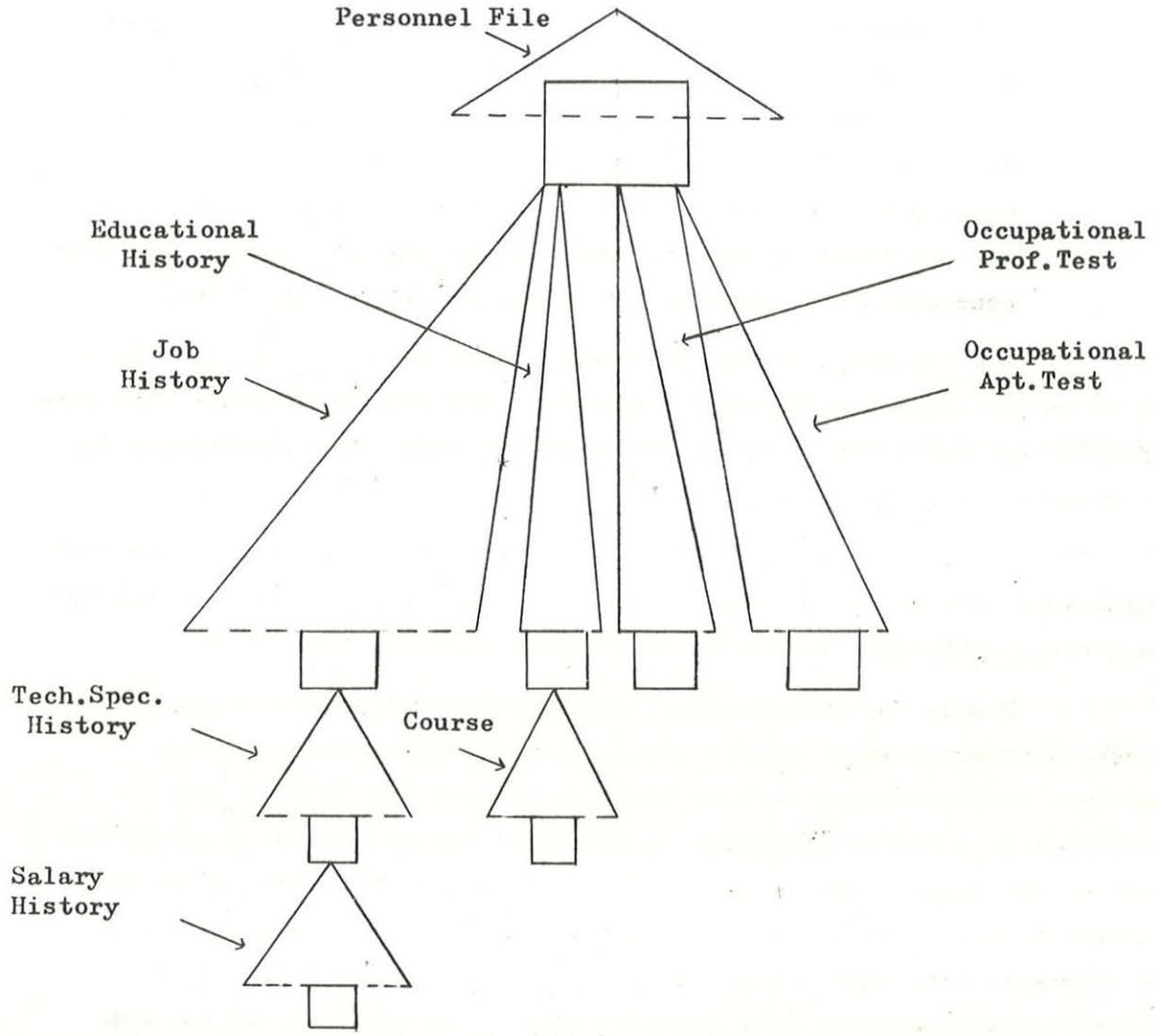

Figure 4: FILE STRUCTURE

2.2 Hardware influence

Historically, the separate files of a system were stored on magnetic tape. The file structure was not unduly complex. The trend however was towards more complex file structures and increasing deterioration in the ability of the system to respond to queries or to changes in the data base was experienced. The limitations of tape oriented systems became acutely apparent and, in the main, may be summarized as follows:

Access to any particular item in a record or a particular record is slow, determined by linear tape movement speed. The volume of swiftly accessible data is low, which also results in slow updating of an item. As a consequence it is necessary to batch both queries and updates and run them during a complete pass of the file. In turn, this has meant that the user cannot query the file directly when he wants to, but must query via an operator when the batch is run. He is not, therefore, getting his results 'at the right time'. Furthermore, updates are not added to the file immediately they are known. We must consequently be content with querying out of date files.

The treatment of the information base as a collection of files is in fact a figment since most queries or 'application programs' are confined to one file only. In an active environment, much information is inevitably duplicated in different files.

The queries to these files must be prescribed and file structures engineered to respond to them. Very little ability to pose new 'unseen' queries is evidenced, within a satisfactory response time.

Mostly these liabilities were encountered on second generation machines although some clearly remain on any tape oriented system. However, there was then a clear division between 'scientific' and 'commercial' modes of operation in hardware, implementation language and ways of thinking. The advent of third generation machines led to the fusion of these differences, at least in hardware and potentially also in language. It also brought with it the direct access devices and multi-access capability from remote points. In addition, we now can look to optical character recognition devices to simplify collection and input of data.

2.3 Recent developments

As a consequence of these hardware advances, large volumes of data (including text) can now be stored with swift access to individual items. Many users can simultaneously obtain direct access at that point in time when they need the information. Remote access points can feed and update data to maintain up-to-date files and to participate fully in the system.

This has thrown into sharp relief the need for high-level

facilities specifically geared to a fast access, high volume, dynamic information system. The emergence of generalized information systems to provide such facilities has been seen in recent years. To some extent, some form of general system has been in use for ten years or so, but outside of military applications, has found only sporadic use. These do not compare with the technical complexity and scope of the facilities of the newly emerging systems. Some of these general systems provide only teleprocessing monitor facilities, or added data management facilities to the standard operating system. Some provide full data base creation and maintenance capability lacking only the query language of a complete system. In figure 5, some of the main systems, currently under development, are given. Some are not yet available, others only recently released. The main advantage of these schemes is that they reduce drastically the time required to construct an information system and to provide the means to change the system in step with the changing environment. In addition, the facilities provide in principle for much more easily prepared queries, widening the accessibility of data base information to an increasingly large group of users, hence bridging the 'technological gap'.

IBM	IMS
	FASTER
	MIS
	GIS
GE	IDS
SDC	TDMS
AUERBACH	DM - 1
INFORMATICS	MARK IV
GENERAL MOTORS	APL

Figure 5: CURRENT INFORMATION SYSTEMS

3. Technical problems in new developments

3.1 Efficiency

The establishment of a data-base to serve a large community of users, some of them simultaneously, will inevitably result in a much larger volume of data to be searched. The development of a good organization of the data and the establishment of search strategies efficiently

to serve the collective need is a much more difficult problem than has been previously encountered. In a very lucid exposition, Dodd (1969) has given an introduction to various methods of basic organization, together with a critical review. He divides them into three types:

1. Sequential organization;
2. Random organization;

and 3. List organization.

Sequential organization essentially mirrors the tape oriented practice discussed earlier and cannot provide the flexibility required for efficient response in any but the simpler systems. Random organization is primarily geared to an indexed set of records and can be harnessed to the hierarchical structure of a data base with reasonable efficiency. A list organization, wherein sub-structures are located by pointers to other physical parts of the store, can also be made efficient for hierarchical structures. The problem of maintaining these pointers after an updating transaction is however most acute. The distribution of data throughout the storage medium must be carefully controlled if seeking along list chains is not to become very time consuming. This whole problem of efficient organization is a most complex one and cannot be discussed further here.

3.2 Redundancy

Although the ability to construct larger integrated files of a much more complex structure has eliminated much of the duplication of information found in a collection of conventional files, the inherent nature of the hierarchy - and most general systems work on the hierarchy as their basis - can still result in considerable duplication. For example, in a supplier-part file, if we establish a structure showing the parts provided by each supplier, as a repeating group, then each occurrence of a particular part in a supplier record must necessitate a repeat of all non-local part information - e.g. 'who uses it', 'what it is used for', etc. Structuring the other way round with 'supplier' subordinate to 'part' yields redundancy of 'supplier' data. One solution is to establish a separate record of non-local data for each such element and to place a reference pointer (direct or indirect) to it in each main record. However this can result in an unacceptable increase in search time or reduce to the problem of efficiently organizing lists - the problem already mentioned in 3.1.

3.3 Data-independence

Amongst the community of users of a data-base it is unlikely that any one person will be aware of the full content of the data-base, nor need he be. Each user is only concerned with that part of the data-base he uses. However, updating transactions are generated at many points by many people and, it must be assumed, will be changing the data base continuously. In particular, it may be necessary to change the structure of the data-base and its organization to better serve the needs of the user community in the environment of the now very different data base. To preserve the integrity of user query programs is very difficult under these changing conditions. In fact, the query languages of today are organization and structure dependent instead of simply content dependent. None of the general information systems mentioned earlier have any significant capability to combat this problem. There is thus a pressing need to develop query languages dependent only on the content of the information base. This is the problem of 'data independence'. It will not therefore be possible, in general, for users to take advantage of the information organization to formulate his query. Some sacrifice in response time is thus inevitable if data-independence in this sense is to be achieved.

3.4 Security and integrity

The centralization of information and the high activity encountered in processing it by many people, throws into sharp focus the need for protective safeguards. It is much more difficult cheaply to ensure that the information base can be swiftly resurrected in the event of destruction. There is much more likelihood of the integrity of the data being threatened in complex structures, especially if some form of list pointer organization is abundant. In this case, one erroneous pointer can result in loss of a large segment of data.

Even more acute is the problem of the security of information ensuring that each user has access to see or change only that part of the information base authorized to him. Preventing or permitting access to the actual data in a data base presents no difficulties in principle, but requires some investigation to prescribe a practical, efficient and effective solution. Two questions of principle arise however when we consider derived information. Firstly, suppose we wish to authorize access to the result of applying a computing procedure to forbidden data. The details of the computing procedure

may not be known and may, in fact, be forbidden. However, the purpose of the procedure must be known if the user is to make valid use of it. The question now arises as to how we can ensure that the user cannot develop an inverse procedure to deduce the forbidden data from the result. In general, it is not possible but in particular cases, serious hazard results.

The second problem revolves around preventing access to the result of applying a computing procedure to a set of permitted data. Again, the procedure must be known in principle, if not in detail. There is no safeguard preventing a user from applying the procedure to the known data to obtain the forbidden result outside of the system. These two problems illustrate the severe logical problems to be faced in prescribing effective security facilities for a large information system.

4. Conclusion

We have introduced the concepts and terminology of information systems and described typical file systems in use on magnetic tape systems. The impact of direct access devices on information systems has been mentioned, leading to the generalized facilities coming on the market today. Some of the problems blocking the way to further progress have been mentioned. The reader is referred to the bibliography for a fuller treatment of all of these developments.

What are the likely developments in information systems technology in the future?

It seems inevitable that data bases will become much larger incorporating higher volumes of both text and numeric data. Increasingly large numbers of users will enter the system from a larger number of remote points. To cope efficiently, complex file organizations based upon some combination of random and list structures must be evolved, closely mirroring the access demand of query traffic. The profile of query traffic will change through time and, to meet the change, the data-base will have to re-organize adaptively in keeping with it.

The distinguishable types of information system will ultimately disappear, providing a deductive capability in the query language together with the ability to communicate in a subset of natural English. Future information systems will be operating in a computer network providing massive computer power, if required, over a far-flung information 'catch-pot' area. The trend towards the so-called 'total program environment' will be seen, wherein any computer program will obtain

its input data as the result of a query to the system and return its output data as an update to the system. Passing of output data from one program as part of the input data to another can easily be effected. Discrepancies in form are immediately resolvable using the information processing facilities of the system. All programs are thus part of the information system.

How many of these developments are likely in the near future?

One fact remains clear. Recent hardware developments are continuing to outstrip the ability of software fully to exploit them. One serious obstacle encountered in planning the future for information systems research is the lack of any acceptance of an adequate theoretical framework. Without the fabric within which to compare, measure and assess, progress must be slower and fashioned by the ingenuity of the experimental scientist.

The potential impact of information systems technology upon society is immense. The functions of management, control and planning in the large and complex systems of today are becoming more intractable each year. This is true not only in commerce and industry, but also in central and local government, in universities and hospitals. To keep pace with this increasing demand, the advancement of information systems technology becomes ever more pressing.

A SHORT BIBLIOGRAPHY

1. Bachman, C.W. (1965) Introduction to Integrated Data Store, G.E. Computer Department, C.P.B. 1048, April 1965.

2. Bachman, C.W. (1966) On a Generalized Language for File Organization and Manipulation. Comm. ACM, 9 (Mar. 1966) 225-226.

3. Barnum, A.R. (1965) Reliability Central Data Management System. In: Rubinoff, M. (ed.) Toward a National Information System; Second Annual National Colloquium on Information Retrieval, April 23-24, 1965, Philadelphia, Pennsylvania. Sponsored by Special Interest Group on Information Retrieval, Association for Computing Machinery; Moore School of Electrical Engineering, Univ. of Pennsylvania; Delaware Valley Chapter, American Documentation Institute, and the Delaware Valley Chapter, Association for Computing Machinery. Spartan Books, Washington, and Macmillan, London, 1965, p.45-61.

4. Bleier, R.E. (1967) Treating Hierarchical Data Structures in the SDC Time-Shared Data Management System (TDMS). System Development Corporation, 2500 Colorado Ave., Santa Monica, California. SP-2750 February 15, 1967.

5. Bobrow, D.G. (1966) Problems in Natural Language Communication with Computers. Bolt Beranek and Newman, Inc., Cambridge, Mass., Aug. 1966, 19 p. (Report no. Scientific-5, BBN-1439) AFCRL 66-620 (AD-639 323).

6. Bryant, J.H. & Parlan Semple, jr. (1966) GIS and File Management. In: National Conference of the Association for Computing Machinery, 21st, Los Angeles, Calif., 30 August - 1 September 1966. Proceedings, p.97-107 (A.C.M. Publication P-66).

7. Climenson, W.D. (1966) File Organization and Search Techniques. In: Cuadra, C.A. (ed.) Annual Review of Information Science and Technology, (American Documentation Institute. Annual Review series, vol. 1). Interscience Publishers, New York, 1966, p. 107-135.

8. Connors, T.B. (1966) ADAM--A Generalized Data Management System In: American Federation of Information Processing Societies. AFIPS Conference Proceedings, vol. 28; 1966 Spring Joint Computer Conference. Spartan Books, Washington, D.C., 1966, p. 193-203.

9. Dodd, G.G. (1966) APL — A Language for Associative Data Handling in PL/1. In: American Federation of Information Processing Societies. AFIPS Conference Proceedings, vol. 29; 1966 Fall Joint Computer Conference, November 7-10, San Francisco, Calif. Spartan Books, Washington, D.C., 1966, p. 677-684.

10. Feldman, J.A. (1965) Aspects of Associative Processing, Tech. Note 1965-13, ESD-TDR-65-65, Lincoln Lab., Massachusetts Institute of Technology, Lexington, Mass., 21 April 1965, 47 pp. (AD-614 634).

11. Green, R.S., J. Minker, & W.E. Shindle. (1966) Analysis of Small Associative Memories for Data Storage and Retrieval Systems. Vol. 1: Management Report. Final report, Oct. 1964 - Sept. 1965. Auerbach Corp., Philadelphia, Pa., July 1966, 116 p. (Report no. 1231-TR-2-Vol-1) RADC TR-65-397-Vol 1 (AD-489 660).

12. Green, R.S., J. Minker, & W.E. Shindle. (1966) Analysis of Small Associative Memories for Data Storage and Retrieval Systems. Vol. 2: Management Report. Final report, Oct. 1964 - Sept. 1965. Auerbach Corp., Philadelphia, Pa., July 1966, 470 p. (Report no. 1231-TR-2-Vol-2) RADC TR-65-397-Vol 2 (AD-489 661).

13. International Business Machines Corporation. (1965) Generalized Information System Application Description. IBM Corp., Technical Publications Dept., White Plains, N.Y., 1965, 42 p.

14. Kasher, A. (1966) Data-Retrieval by Computer; a Critical Survey. Hebrew Univ., Jerusalem, Israel, Jan. 1966, 72 p. (Technical Report No. 22 to Office of Naval Research, Information Systems Branch). (AD-631 748).

15. Kellogg, C.H. (1966) An Approach to the On-Line Interrogation of Structured Files of facts Using Natural Language. System Development Corp., Santa Monica, Calif., 29 Apr. 1966, 86 p. (SP-2431/000/00).

16. Langefors, B. (1966) Theoretical Analysis of Information Systems. Vols. 1 & 2: Studentlitteratur Lund, Akademisk Foslag, Kobenhaven.

17. Levien, R.E. (1966) Relational Data File II: Implementation. Rand Corp., Santa Monica, Calif., July 1966, 24 p. (P-3411).

18. Mann, W.C., & P.A. Jensen. (1966) A Data Structure for Directed Graphs in Man-Machine Processing. Computer Command and Control Co., Washington, D.C., 20 June 1966, 76 p. Report no. 77-206-1. (AD-636 251).

19. Prywes, N.S. & H.J. Gray. (1962) The Multi-List System for Real-Time Storage and Retrieval. In: International Federation for Information Processing. Information Processing 1962; Proceedings of IFIP Congress, Munich, 27 August - 1 September 1962. North-Holland Publ. Co., Amsterdam, 1963, p. 273-279.

20. Rocchio, J.J., Jr. (1966) Document Retrieval Systems — Optimization and Evaluation. Thesis (Ph. D.) Harvard Univ., Cambridge, Mass., Mar. 1966, 1 vol. (Harvard Univ., Computation Lab. Information Storage and Retrieval. Scientific Report no. ISR-10 to the National Science Foundation).

21. Rovner, P.D. (1966) An Investigation into Paging a Software-Simulated Associative Memory System. Office of Secretary of Defense, Advanced Research Projects Agency. Document No. 40.10.90, Contract SD-185, January 18, 1966.

22. Salton, G. (1966) Data Manipulation and Programming Problems in Automatic Information Retrieval. Comm. ACM, 9 (Mar. 1966) 204-210.

23. Salton, G. (1966) The Representation of Data Structures in Information Systems. In: Kalenich, W.A. (ed.) Information Processing 1965. Proceedings of IFIP Congress 65. Vol. 2: (Addresses at the opening and closing sessions, summaries of the symposium sessions, and reports of panel discussions.) Organized by the International Federation for Information Processing, New York City, May 24-29, 1965. Spartan Books, Washington, D.C.,; Macmillan and Co., Ltd., London, 1966, p.345-347.

24. Salton, G. (1966) The SMART System — Retrieval Results and Fugure Plans. In: Cornell University. Department of Computer Science. Information Storage and Retrieval. Scientific report no. ISR-11 to the National Science Foundation. Gerard Salton, Projector Director. Ithaca, N.Y., June 1966, Sec. 1, 9p.

25. Salton, G. & E.H. Sussenguth, Jr.(1964) Some Flexible Information Retrieval Systems Using Structure Matching Procedures. In: American Federation of Information Processing Societies. AFIPS Conference Proceedings, Vol. 25; 1964 Spring Joint Computer Conference, Washington, D.C., April 1964. Spartan Books, Baltimore, Md., 1964, p. 587-597.

26. Simmons, R.F. (1965) Answering English Questions by Computer: A Survey. System Development Corp., Santa Monica, Calif., Apr. 1964. (SP-1556) Slightly revised version published in: Comm. ACM, 8 (Jan. 1965) 53-70.

27. Thompson, F.B. (1966) English for the Computer. In: American Federation of Information Processing Societies. AFIPS Conference Proceedings, vol. 29; 1966 Fall Joint Computer Conference, November 7-10, San Francisco, Calif. Spartan Books, Washington, D.C., 1966 p.349-356.

28. Vorhaus, A.F., & R.D. Wills. (1967) The Time-Shared Data Management System: A New Approach to Data Management. System Development Corporation, 2500 Colorado Ave., Santa Monica, California. SP-2747. February 13, 1967.