SCISPACE
formerly Typeset

Open access • Journal Article • DOI:10.1086/620619

The University of Chicago Library Data Management System — Source link

Charles T. Payne, Rob McGee, Helen F. Schmierer, Howard S. Harris

Published on: 01 Jan 1977 - The Library Quarterly (University of Chicago Press)

Topics: Cataloging, Library catalog, Resource Description and Access, Digital library and Data management

Related papers:

- Library Information System Time-Sharing (LISTS) Project. Final Report.
- A library management information system in a multi-campus environment
- Computer-derived management information in a special library
- On-Line Computer Techniques in Shared Cataloging.
- LOLA; Library On-Line Acquisitions Sub-System, Washington State University.

Share this paper:

View more about this paper here: https://typeset.io/papers/the-university-of-chicago-library-data-management-system-sv2ea3pz7j

CHARLES T. PAYNE
Systems Development Librarian
University Library
University of Chicago
Chicago, Illinois

The University of Chicago Library Data Management System

This paper describes the computerized library data system designed and built at the University of Chicago. The project is supported by grants from the Council on Library Resources and the National Endowment for the Humanities and is an extension of an earlier project at Chicago supported by the National Science Foundation. The Chicago system is large and complex, and can be viewed from a number of different aspects, all important to an understanding of the total. The Chicago system is a large data base system; it is also a library data processing system, a data management system, an access system, and a data communication system. It has both hardware and software components, and it makes use of two computers: one large, one small. The University of Chicago Computation Center facility provides the main computing power and data base management. A smaller, front-end computer handles the library's network of forty to fifty terminals and provides a high-speed interface to the Computation Center.

We will look at the Chicago system as a data processing system, as a large data base system, and as a data access system, and also look at the hardware configuration and implementation. First, however, it is appropriate to review briefly the development of the Chicago system.

This is a second-generation project—i.e., one whose staff has had previous experience and has learned the hard way about systems development and implementation. At Chicago, by the late 1960s we had built and were operating a bibliographic data processing system for the library. It allowed data to be input to an in-process file at the time either of ordering or cataloging, accepted input from either MARC tapes or keyboard terminals, and produced printed products for the library. It also produced a large number of machine-readable bibliographic records.[1]

The system has done a lot of production work for the library, handling all Roman-alphabet cataloging and ordering since 1968, and some parts are being incorporated into the new system. The experience with development of this first system, with the operation of it, and with the limitations of it helped shape Chicago's current second-generation project.

In the early part of the current project we undertook extensive work in two areas: (1) to determine and state clearly library data processing requirements, and (2) to develop the computer software capabilities needed to handle library data processing requirements. Most of the exploratory and analytical work is finished and the systems designed. An orderly implementation of the new system is taking place now and will continue through 1974.

Library Data Processing

Herman Fussler, in a Sloan Foundation report, says that in order to improve information access or costs of library processing operations, changes more basic than simple machine replication of existing routines and processes will be required.[2] Experience at Chicago can add that a data processing system which merely prints products for use in library processing operations does not necessarily improve the way that operation is carried out or even affect the efficiency of it. Clearly, almost all library processing is, in fact, data processing, and as long as library data are processed on slips of paper and maintained and accessed in multiple copies in multiple manual files there will be real limits to the efficiency of the operations. In addition, the larger manual files become, the more difficult the update function is to maintain, and the more difficult current information is to find.

From our analysis at Chicago, the most effective model covering library processing and query operations seems to be one where every operation has immediate access to up-to-date information.

Figure 1 presents a conceptual approach to such a model for library data processing. The essential feature of this model is the central Library Data Base, containing bibliographic and operational data. Assume immediate access to data from all operational points, e.g., selection, circulation, cataloging. Data input is a natural part of the work of each operational point and, once input, the data are, in general, available to all points. Updating becomes an automatic part of most processes, and an update in one operation is an update for everyone. To a very large extent the need for replicating data on multiple

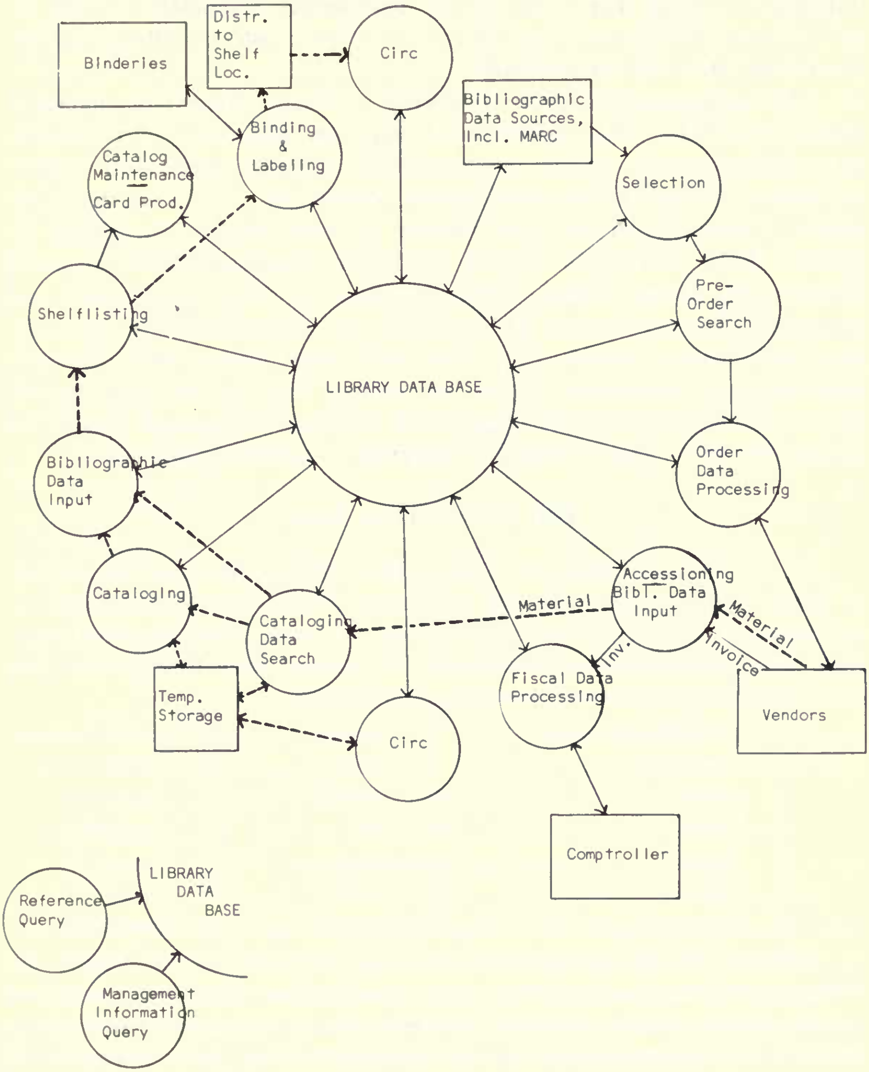

Fig. 1. Conceptual Model of Library Data Processing

pieces of paper is eliminated. Ideally, of course, the total of the library's bibliographic and process-related data would be housed in this single data base including everything that is now in the card catalogs. Practically speaking, we are not quite to that point yet, but the rest of this model is realizable and we are thinking about the card catalogs.

The implementation at the University of Chicago follows this model. It makes a significant computerized library data base available to library processing activities through on-line, interactive CRT terminals scattered throughout the library's technical processing, public service, and circulation areas. The system is sized to handle large volumes of data transfer. The CRTs can quickly display screens of data as needed in processing. Each processing operation has on-line access to the data base for querying, searching, and comparing. Each section has the capability and responsibility for input, update, and quality control of its own data. These library data processing capabilities are incorporated in the new system. The structure and composition of the data base and the design of the data base management system will be discussed next, before turning to the system's hardware configuration.

The Library Data Base

In the Chicago system the Library Data Base is defined as the network of library data—the set of all files of data plus the relations among them.[3] In the examination of library activities and data, the project staff identified those areas where library data are maintained. The Library Data Base includes data about twenty-one such areas, some large, some small:

1.	bibliographic item	8.	invoice	15.	seminar room
2.	authority	9.	fund	16.	action date
3.	order	10.	account	17.	employee
4.	vendor	11.	loan	18.	equipment
5.	donor	12.	patron	19.	person
6.	binder	13.	locker	20.	course
7.	shipment	14.	study	21.	department.[4]

The set of all data for an area comprises a file or files—hence the bibliographic item data file, the order data file, etc. Each file consists of records and records contain data. Data and records may be interrelated, both within a file and among files, and the relationships may be both simple and complex.

Figure 2 presents a simplified, partial picture of the Library Data Base, relating bibliographic data and processing data. The Bibliographic Item File is of central importance in this network of data and is by far the largest file.

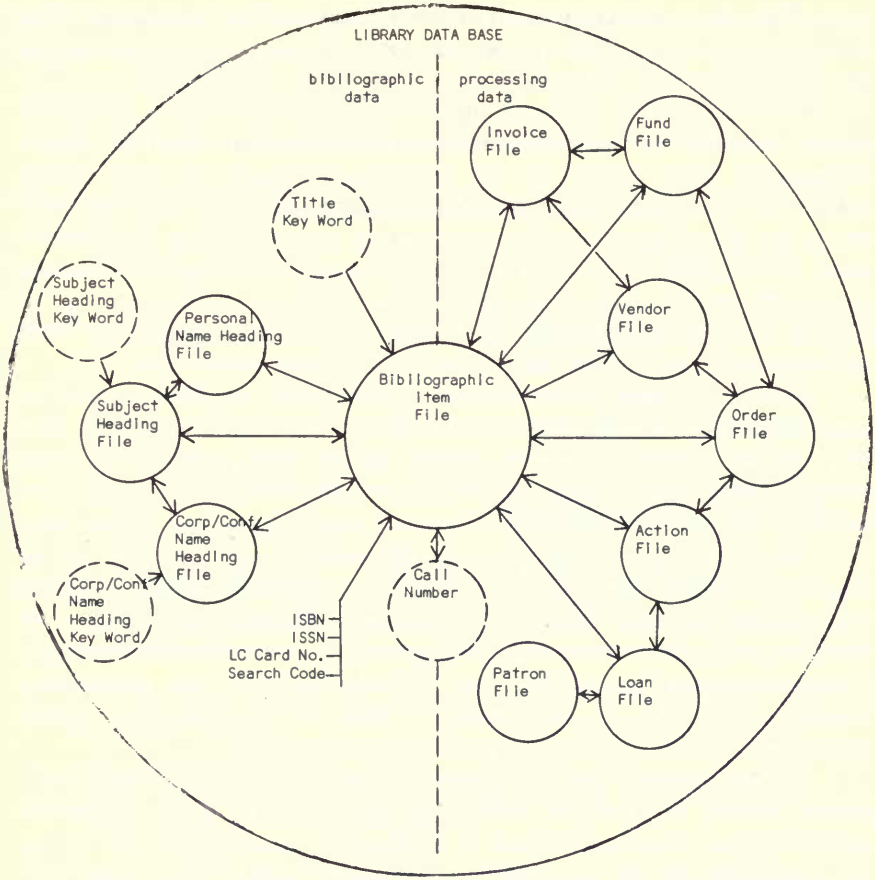

Fig. 2. Library Data Base

Each processing data file by itself tends to be rather straightforward and easy to define. Taken together, however, they require a system capability to handle multiple, interrelated files. In these multiple files there is a minimum of redundant data stored. Instead pointers are used to connect data in various files. In almost every case, for example, the processing files are related to and depend upon the Bibliographic Item File for the bibliographic description data. A record in the Order File has an order number, special instructions, and specific volumes, copies, or pieces wanted, etc., plus pointers to the Vendor File, Fund File, Action File (claim date), and Bibliographic Item File. The

Order File also serves as an order number index to the other related files. Pointers work in both directions so that the order record can be reached via the bibliographic item record, fund record, and other related records.

The bibliographic data portion of the data base contains the Bibliographic Item File and the related set of access and authority files and indexes. Bibliographic item records are not stored as a sequence of unit records. Instead, the data structures are designed to provide maximum access capability and at the same time minimum redundancy in storage of data. The Personal Name Heading File, for example, holds the personal name data for the records in the Bibliographic Item File, serves as an index to it, and acts as the personal name heading authority file. Access to the Bibliographic Item File is also enhanced by browsable Call Number and Key Word index files.

In order to display a complete unit record with both bibliographic and processing data it is necessary to pull together data from a number of files. Few retrievals, however, require all data for an item. Also, for common linkages of data, efficient retrieval sometimes requires redundant storage of data. System tuning includes balancing of retrieval needs against data storage costs.

Even within the bibliographic item file itself, the data structures are not simple unit records. Data here are stored in a quadraplanar structure designed to allow multiple users and uses of a common bibliographic data base and yet maintain relationships such as the holdings of an institution or the catalog of a collection, again with a minimum of redundant data. The quadraplanar structure contains four types of planes, or levels, of data, some with multiple occurrences within a given record in the file (see figure 3).

The bibliographic item file structure is designed to accommodate bibliographic records for multiple collections and for either single or multiple institutions per collection. The set of bibliographic records in the file for any one institution may or may not be a catalog—a catalog being defined as a collection under common authority control. Authority control is an option for each collection, and each collection has its own authority control. The data are arrayed so that all collections share data common to all bibliographic records, and so that each collection shares common authority data. Institution-specific data are maintained per institution.

In the quadraplanar structure, the universal plane holds data that are not variable from one copy of a bibliographic item to another. The collection plane, in each instance, holds descriptive data that do not vary within a given catalog and authority structure. (There may be multiple collection planes representing multiple catalogs in the file, however.) Each institution plane holds data that vary between institutions which belong to the same collection. The copy plane holds data that vary with the physical piece.

Data in the Bibliographic Item File are arrayed in the Quadraplanar Bibliographic Item Data Structure. There are four types of planes in the Quadraplanar Structure. The four types of planes in the structure and the data present in each plane are:

U, the universal plane - contains data common to all collections; includes ISBD(M)data, ISBD(S) data, ISBN, ISSN.

There is exactly one universal plane in the Quadraplanar Bibliographic Item Data Structure.

C, the collection plane - contains data which are collection-dependent and common to all institutions contributing to the collection; includes entries assigned to the universal data in the process of merging an item into a collection--the choice and form of entries, e.g., choice and form of main, added, and subject entries.

For N collections $(C_1,\ldots,C_N)$, there are N collection planes in the Quadraplanar Bibliographic Item Data Structure.

I, the institution plane - contains data which are institution-dependent and copy-independent; includes call number on "cataloging record," local notes, "added" added entries, and copy-independent processing data and series decisions.

For M_i institutions $(I_{i,1},\ldots,I_{i,M_i})$ contributing to collection C_i, there are M_i institution planes in the Quadraplanar Bibliographic Item Data Structure. The total number of institution planes in the structure for N collections, M institutions is:

$$M = \sum_{i=1}^{N} M_i .$$

I^c, the copy plane - contains copy-dependent data per institution; includes call number as on physical piece, and copy-dependent data and series decisions.

There is exactly one copy plane per institution plane in the Quadraplanar Bibliographic Item Data Structure; so for M institutions there are M copy planes in the file.

Fig. 3. Quadraplanar Bibliographic Item Data Structure[4]

In its initial implementation this data base system will be built and operated for a single, large university research library with its own bibliographic and library files. The initial data base at the University of Chicago will consist of Roman-alphabet cataloging since 1968; recent Library of Congress MARC data; partial authority data; and processing data files for circulation, technical processing, and library administration. In regard to processing data, the machine files are intended to replace all manual (paper) files. In regard to bibliographic data, the machine files are designed to include all functions of a true catalog (i.e., control, authorities, and cross references), and replacement of certain departmental and area catalogs is intended as soon as possible. Eventual closing of the general card catalog is foreseen, and this system is designed to replace the catalog.[5]

Data Base Management System

In order to build, maintain, and access a complex, multifile machine data base there needs to be data base management software. A significant analytical and design effort was carried out by the Chicago project staff to produce the system now installed at Chicago. The resulting system, called HERMES, is outlined in figure 4.

A quality of the HERMES design is its general, as opposed to library-specific, approach to data handling and file management. These basic capabilities are provided by a set of software packages, both commercially supplied and locally developed. HERMES has capabilities designed to meet library requirements and also has capabilities sufficient to meet or exceed the needs of most other large file applications within the university. In its implementation at the Computation Center, the basic data and file handling capabilities of HERMES can be shared with other large file applications on campus, thus broadening the base for support of its maintenance.

A commercial on-line control package handles terminals and input and output of data—converting incoming data to a standard form for the applications programs, handling queues and priorities, and converting outgoing data to a form for transmission. The applications programs are specifically designed for each function, but in general, control data processing.

Chicago Access Support Module (CHASM) controls multiple users, multiple files, and security. It also handles interfile relationships, and in this the CHASM design goes considerably beyond most file management software. In the Chicago system, as noted previously, data base is defined as the set of all files plus the relations among them. The data base management software, therefore, needs to provide capabilities for access to and control of any number of interrelated files. Central control of interfile relations promotes uniform integrity of the data base and also removes responsibility for

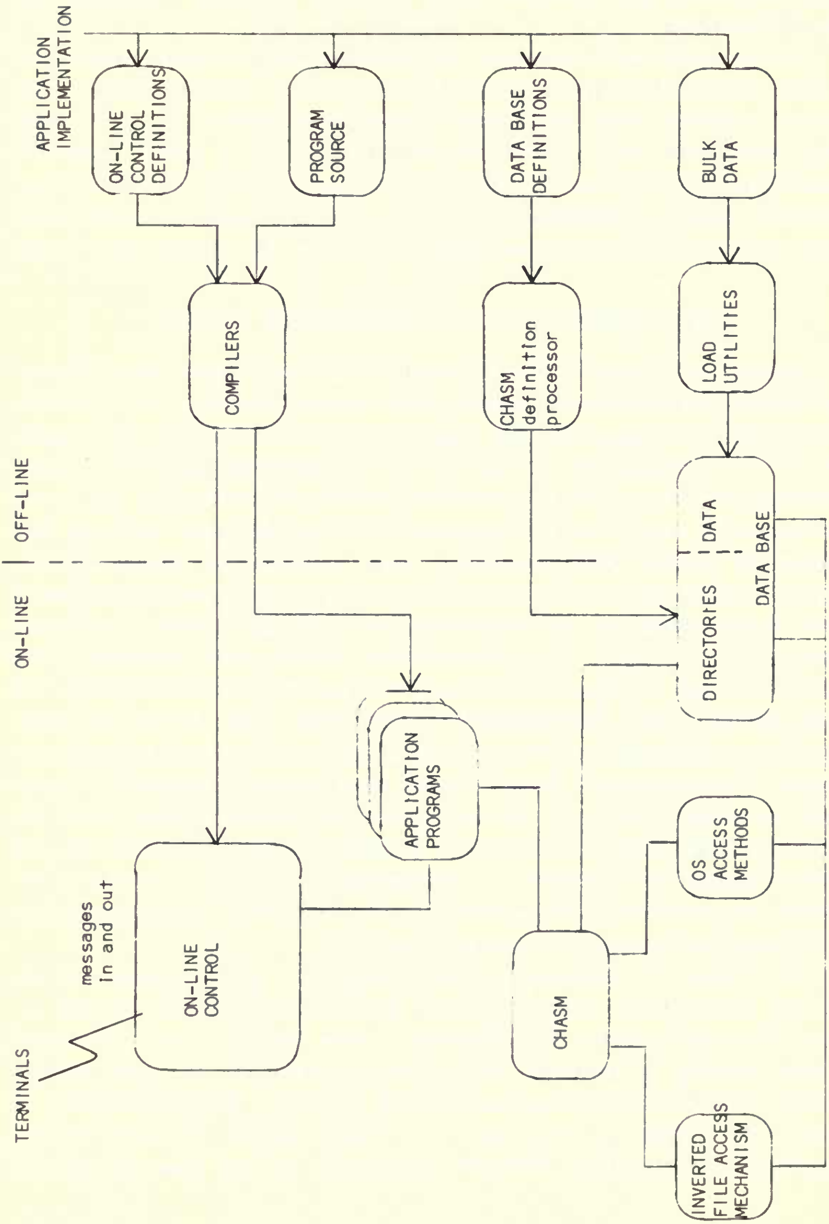

Fig. 4. HERMES On-line Data Base System[6]

maintenance of interfile relations from the applications programs. CHASM uses various access methods including a commercially supplied inverted file access mechanism and IBM/OS access methods, including Sequential and Indexed Sequential.

In the overall software design, the relationships of software components are such that applications programs—the programs for specific library operations—are separated from the physical characteristics (speed, character coding, control characters, etc.) of external equipment, such as CRTs, typewriter terminals, and printers, and from the physical aspects of data storage on disks. Separation permits applications to be programmed at a logical data level, which makes coding, testing, and subsequent maintenance and change much easier.

The Hardware and Communications System

In the early stages of the University of Chicago Library Data Management Project, as the general system design began to emerge, the hardware system design also began to take shape. At first there was no clear requirement for a minicomputer. In fact, it was the university policy that major computing be done at the Computation Center's central facility, and it was also reasonably clear from the beginning that the large data base and data management system would require a large computer for implementation. Later, as detailed library requirements and system design showed the extent to which on-line interaction was desired and the potential number of terminals needed, the advantages of a minicomputer became more apparent. Data transfer rates and the approximate number of terminals were determined, and a study showed that a minicomputer as part of the front end for the system had definite operational and economic advantages in the Chicago environment. Without a minicomputer, several dozen lines to the Computation Center would be needed. The costs of lines, ports, and modems would offset the costs of a single high-speed line and a small computer in the library. We therefore decided to incorporate a minicomputer front end. A diagram of the hardware and communication system as finally established for implementation of the library data system is shown in figure 5.

The Varian 73 computer in this configuration is not, perhaps, a true front-end processor, at least as defined by John T. M. Pryke, who has described front-end systems for IBM users. Pryke specifies that a front-end processor handles all terminals and connects directly to the IBM selector channel, with appropriate front-end control software.[7] At the University of Chicago, the library is only one of a large number of users of the Computation Center's IBM 370/168 system and did not have this option.

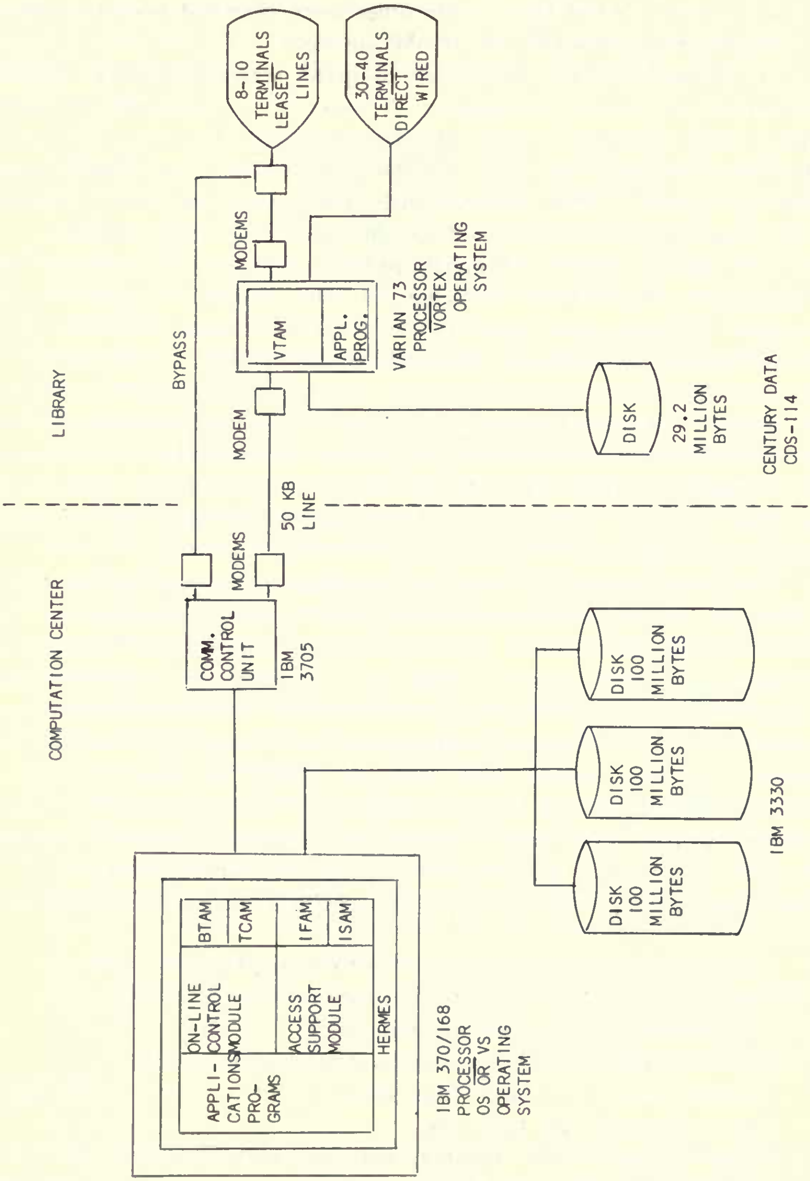

Fig. 5. Communications and Hardware in Use at University of Chicago

Instead, the library processor is connected to the channel by an IBM 3705 Communications Control Unit, as are other users. Standard OS telecommunications control programs (BTAM, TCAM) are used.

In this configuration the Varian computer serves principally as a data concentrator and remote multiplexor. It converts data generated by all the various library terminals to look like data generated by IBM 3270 terminals before transmission to the IBM 370/168 facility, which has software established for support of IBM 3270 terminals. The number and type of terminals in the library system are not, therefore, apparent to the main computer.

Any make or design of terminal can be handled through the system with an appropriate program in the Varian 73. This is true of terminals at the University of Chicago and would also be true of terminals not at the University of Chicago, which allows potentially great flexibility in communications with other libraries or network systems. Another advantage of this configuration is that it provides a measure of back-up. If the main computer is down the Varian 73 provides a continuity of service at the terminals. In particular, the circulation function demands uninterrupted service about 18 hours per day. The Varian 73 can, at least, check patron privileges, provide charge evidence, and log data for file updates in the main computer.

Various definitions for minicomputer exist, some suggesting upper limits on the size (or cost) of a machine that can be called a minicomputer. The Varian 73 falls above most such limits, and processors in its range are sometimes called midicomputers rather than minicomputers. Such distinctions are probably not important. What is important is that every machine be of adequate size for its processing requirements. At Chicago, specifications for a front-end computer system were drawn up taking into account projected terminal transactions and data transmission loads, peakload as well as average.[8] Specifications were submitted for bid for a front-end computer system to serve as a data concentrator and high-speed interface to the Computation Center's IBM 370/168. The specifications include the following:

1. processor speed—sufficient to drive the estimated mix of terminals;
2. disk storage—sufficiently large to include software plus certain files;
3. tape drives—two, to log all transactions;
4. console—operator communications and programming tool;
5. communications interfaces—high speed to the Computation Center, a mix of speeds for the terminals;
6. system software—disk oriented, with assemblers or compilers and a communications package; and
7. service—to be locally available and reliable.

Notice that these specifications are not simply for a computer but for a complete system including the communications interfaces.

The specifications were sent to eleven manufacturers and suppliers, and resulted in nine proposals. These responses were evaluated according to the following criteria:

1. Reputation of manufacturer/supplier—This criterion is the most subjective, but in terms of our requirements, is the most important. The hardware must be widely available and supported over the life of the system, which suggests the importance of dealing with an established vendor. The product must be currently demonstrable and of proven reliability.
2. Maintenance—Maintenance support must be available locally and should be of proven reliability.
3. Communications hardware architecture—The major task of the minicomputer system is that of driving approximately forty-eight lines with various transmission characteristics. The communications hardware must be capable of handling a variety of device types and communications characteristics and of processing the anticipated load, and have sufficient reserve power for expansion. Careful attention must be paid to whether the communications hardware controls transmission on a character-by-character basis (programmed I/O) or on a message basis (direct memory access).
4. Communications software support—In conjunction with the communications hardware, an extremely desirable feature is the availability of communications software.
5. Peripheral devices—Because the front-end computer system provides back-up when the main computer is down, a heavy burden is placed on the peripheral devices, most notably on the disk drives which hold several processing files. The mechanical components of the system have the least reliability. Therefore, the disk and tape drives proposed must be devices previously installed and of high reliability.
6. CPU architecture—The state of the art in processor design has reached a point that many fast, reliable minicomputers are on the market. Therefore, CPU to CPU comparisons should be minimized as much as possible.[9]

The Varian proposal was judged best at meeting these criteria and meeting the hardware and software specifications. The Varian system has a 32K Varian 73 CPU. The system has a sixty-four line Data Communications

Multiplexor for all line interfaces and is designed to operate with a throughput of 60,000 characters per second. The DCM is a direct memory access device with a core table for each line on the device providing line dependent information such as buffer address and byte count. The DCM is supported by VTAM, the communications portion of Varian's operating system VORTEX. Various line adapters including synchronous and asynchronous may be tied to the DCM. The character size may be 6, 7, or 8 bit. Parity checking is provided. All lines may be either full- or half-duplex.

Implementation

The complex, multiple-link communications and hardware system (figure 5) is currently being implemented at the University of Chicago. We can report that such systems do not get into operation overnight. Such systems may require several months to install, to debug, and to test all communications links in all modes of operation.

We are, at one time, testing terminals and programming screens for them; installing communications cables from terminal locations to the Varian site; programming the Varian; testing the data links through the Varian, the modems, the high-speed cable, and the IBM 3705 to the IBM 370/168; enhancing CHASM; developing applications programs; and planning some major file conversions.

It is a busy and difficult period but things are starting to fit. Already some acquisitions and cataloging operations are functioning on-line. The communications system will soon be operational, and the large bibliographic item file (250,000 records) is to be on-line for use by June 1974. By October 1974 the circulation system is to be in operation, and other technical processing applications will be installed by the end of 1974.

Our experience with the development of the University of Chicago Library Data Management System prompts a particular observation. Librarians have done fairly well in the last several years at becoming comfortable with the idea of computers. Computer systems analysts and programmers are fairly common now on library staffs. But another acquaintance probably needs to be made. The University of Chicago system described above is an electronic communication system. Each of the hardware pieces mentioned contains electronic circuitry. Few librarians (and not many more programmers) can read an electronic circuit board, or repair it, or determine the exact point of failure in a system.

Such electronics expertise is needed, and it probably cannot be supplied effectively by service or maintenance representatives of supplier companies. It is difficult for the library to coordinate the technical representatives of four to six different companies when the library does not have the ability to determine whose equipment is at fault. The need for local electronics expertise is even more pronounced if one desires to alter or modify hardware. At the University of Chicago the library has established a working relationship with the staff of the electronics shop of the Enrico Fermi Institute, and their help has proven to be vital to implementation of the library hardware and communications system. We recommend that any library planning to establish any kind of electronic data system try to find an electronics laboratory or shop somewhere nearby and start to get acquainted.

REFERENCES

1. Chicago. University. Library. *The University of Chicago Library Bibliographic Data Processing System: Documentation and Report as of October 31, 1969*. By Charles T. Payne, *et al.* Chicago, University of Chicago Library, 1970; and ————. *Supplement*. By Charles T. Payne and Robert S. McGee. Chicago, University of Chicago Library, 1973.

2. Fussler, Herman H. *Research Libraries and Technology*. Chicago, University of Chicago Press, 1973, p. 11.

3. Chicago. University. *Library Data Management Project. SuperCHASM Design* (Document no. 87). By Emil Baciu, *et al.* Chicago, University of Chicago Library, 1973, p. 1.

4. ————. *Quadraplanar Bibliographic Item Structure* (Document no. 89). By Jane Jodeit. Chicago, University of Chicago Library, 1973, p. 5.

5. ————. *Library Data Management; Feasibility and Design. Report to the Council on Library Resources and the National Endowment for the Humanities*. Chicago, University of Chicago Library, 1972.

6. ————. *An Introduction to HERMES, An On-Line Data Base Management System* (Document no. 97). By Randall M. Lee. Chicago, University of Chicago Library, 1973, p. 8.

7. Pryke, John T. M. "A Front-End Primer for IBM Users," *Datamation*, 19:46-50, April 1973.

8. Chicago. University. Library Data Management Project. *Specifications for a Front End Computer System and Circulation and Technical Processing Terminals* (Document no. 57). By Randall M. Lee. Chicago, University of Chicago Library, 1972.

9. ————. *Selection of a Front End Computer System for Regenstein Library* (Document no. 70). By Randall M. Lee, *et al.* Chicago, University of Chicago Library, 1973, pp. 6-7.