

Design Principles for a Comprehensive Library System

Tamer ULUAKAR, Anton R. PIERCE, and Vinod CHACHRA:
Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

This paper describes a project that takes a step-by-step or incremental approach to the development of an online comprehensive system running on a dedicated computer. The described design paid particular attention to present and predicted capabilities in computing as well as to trends in library automation. The resultant system is now in its second of three releases, having tied together circulation control, catalog access, and serial holdings.

PERSPECTIVE

The use of computers in libraries is no longer a speculative venture for the daring few. Rather, library automation has become the accepted prerequisite for effective library service. The question faced is not "if," but rather "how" and "when." The reasons for this evolution are diverse, but fundamental is the recognition of online computer processing as the most effective means of simultaneously handling inventory control, information retrieval, and networking of large, complex, and volatile stores of data. Most areas of current library practice could now benefit from effective computer-based control. Mature and proven systems exist for cataloging, circulation, serials control, acquisitions, catalog access, and "reader guidance"; the latter by virtue of online literature searching facilities such as DIALOG, MEDLARS, or BRS. The challenge is to find or develop an optimal mix of capabilities.

Two common limitations from which library automation projects suffer are the use of nonstandardized, incomplete records and the lack of functional integration of different tasks. In most cases these limitations are due to historic circumstances. The pioneering systems — say, those online systems introduced between 1967 and 1975 — had to conserve carefully the available computing resources. A decade ago it was unthinkable for any library to store a million MARC records online. Mass

storage costs alone precluded that option. To best realize the benefits of automation, short records, usually of fixed length, were employed.

There is little question that systems based on short records were helpful to their users. However, one characteristic of these systems was their proliferation within a particular library. After the first system was shown to be a success, it became compelling to try another. The problem was that these separate systems were usually not communicating directly with each other because of limitations imposed by program complexity and load on available resources.

Thus, the use of incomplete records breeds isolated, noncommunicating systems. However, system users have come to demand that all relevant data be available *at* a single terminal *from* a single system. It is not enough to know that a particular title is due back in twenty-five days; the user must also know that copy two has just been received, and that copy three is expected to arrive from the vendor in one week. That is, the functions of catalog access, circulation, and acquisitions must be brought together at a single place — the user's terminal. And while the importance of functional integration has been recognized for some time, only a very few report successful implementations.^{1,2} The Kafkaesque alternative to functional integration becomes the library that has been "well computerized" but where the librarian must use five different terminals, one for each task.

As computer-based systems have grown to maturity, increasing stress has been placed on standardization. In library automation the measure of standardization is wide-scale use of the MARC formats for documents and authorities; the use of bibliographic "registry" entries such as ISBN, ISSN, or CODEN; the use of standard bibliographic description; and so forth. However, the application of common languages and standardized protocols, data description, and definition has been less pervasive. We find many applications that eschew use of the common high-level languages, database management systems, and standard "off-the-shelf" or general-purpose hardware.

The emergence of powerful and easy-to-use database management systems, the spectacular price reductions in hardware, and the concomitant, and equally spectacular, improvements in system capabilities have made it clear that it is practical to think ambitiously. Perhaps the major articulation of these developments has been the pervasive shift from a central computer shared with nonlibrary users to the utilization of dedicated minicomputers.³

Our analysis of the requirements of a comprehensive system led to recognition of the key role played by serials in research libraries. Serials form the most critical factor in automating library service because of the complexity of their bibliographic, order, and inventory records, and because of their importance to research.⁴ A fundamental error in designing a comprehensive library system would involve focusing on the require-

ments of monographs and/or other "one-shot" forms of the literature. The reason is, simply, that monographs and other such publications can be treated as an easy limiting case of a continuing set of publications. This observation is borne out by Christoffersson, who reports an application that extends the idea of seriality and develops a means to provide useful control and access to all classes of material.⁵

DESIGN PHILOSOPHY

The concerns outlined above mean that a viable library system should meet the following design criteria:

Functional integration. Functional integration is simply the ability to conduct all appropriate inquiries, updates, and transactions on any terminal. This envisages a cradle-to-grave system wherein a title is ordered, has its bibliographic record added to the database, is received and paid, has its bibliographic record adjusted to match the piece, is bound, found by author, title, subject, series, etc., charged out, and, alas, flagged as missing. In this way a terminal linked to the system will be a one-stop place to conduct all the business associated with a particular title, subject, series, order, claim, vendor, or borrower.

Completeness of data. If the system is to be functionally integrated, it is clear that it must carry the data required to support all functions. In particular, data completeness is required to satisfy the access and control functions. Consider, for example, the problems associated with the cataloging function. A book is frequently known by several titles or authors. Creating these additional access points is a large portion of the cataloger's responsibility. Only systems that allow the user access to these additional entries utilize the effort spent in building the catalog record. Such system capabilities must be present to allow the labor-intensive card catalog to be closed and, more important, to allow maintenance of the catalog *within* the system.

Use of standardized data and networking. In an excellent article, Silberstein reminds us that, in general, the primary rationale for adhering to standards is interchangeability.⁶ We give great importance to being able to project our data to whatever systems may develop in the future. We believe this consideration is of the highest priority because, fundamentally, the only thing that will be preserved into the future is the data itself.*

Without interchangeability of data, sharing of resources is impossible. Data interchangeability is, of course, a basic assumption that has been made in speculation concerning the national bibliographic network⁷ developing from the bibliographic utilities—notably, OCLC, Inc., the Research Libraries Group's RLIN facility, the Washington Library Network, and the University of Toronto's UTLAS facility. Today, nearly all

*This state of affairs seems to be true for all computer-based systems because their lifetime is, typically, no greater than ten years.

research libraries participate in some utility. While their participation is primarily directed to utilization of the cataloging support services, we find an increasing amount of interest and use of additional capabilities, notably interlibrary loan. We expect a steady and continual growth of these library networking capabilities.

However, networking is *not* problem free. Perhaps the biggest single problem in using the network is the misalignment between the record as found on the bibliographic database and the requirements of individual libraries. While such variability between the resource database record and the user's needed version is well understood,⁸ the local library frequently has a difficult time adjusting records to meet local needs. One example is OCLC's inability to "remember" in the online database a particular library's version of a record. Another example is the CONSER project's practice of "locking" very dynamic records as soon as they are authenticated. This locking frequently means that required updates cannot be made and users cannot share with one another corrections to the base record. After locking, each must, independently, go about bringing the record up to date. Thus, as Roughton notes, "the next library to call up the record loses the benefit of the previous library's work."⁹ This inhospitable state of affairs forces individual libraries to maintain their own records if they wish to change bibliographic records after initial entry.

The problem of local adjustment of bibliographic records in no way conflicts with the goal of standardized bibliographic data. Standardized data provides a quick means of delivering an intelligible package to a variety of users who will adapt the package to meet their particular needs. Standardization does not mean making adaptation inefficient or more costly than it need be; rather, standards provide a framework around which the details are filled in. These observations on standardized data formats imply that the library's data must be based on MARC records for books, serials, authorities, etc.; and on the ANSI standards for summary serials holdings notation, book numbers, library addresses, and so forth.

Microscopic data description. At this point, system administrators face a fundamental problem—many of the library's important records have *no* standard format. The most conspicuous example involves the notation for *detailed* serials holdings.¹⁰ The only alternative one has when trying to build a system without standardized formats is to rely on "microscopic" description. That is, each and every distinct type of data element that makes up (or can make up) a field in a record must be accounted for and uniquely tagged. In this way, whatever standard format is ultimately set, it will be possible, in principle, to assemble *by algorithm* the data elements into an arrangement that will be in conformity with the standard. Only if the library is using microscopic data description will the library be able to maintain its independence of particular lines

of hardware or software. We are convinced that the use of untagged, free-form input will, in the long run, spell disaster.

Use of general purpose hardware and software. Many strategies in dealing with library automation involve redesigning standard hardware or software. For example, one vendor has reported an interesting design of mass storage units that improved access time.¹¹ We feel that future applications should, as much as possible, steer clear of such customized implementations because the standard capabilities of most affordable systems allow sufficient processing power and storage economies even if these capabilities are suboptimal for a particular application. The use of general-purpose hardware and system software promotes system sharing between different installations. Moreover, an application based on general-purpose hardware and system software will be easier to maintain and far less vulnerable to changes in personnel. For turnkey installations, the greater the degree of use of general-purpose hardware and software, the better shielded will the installation be against changes in product line or the vendor's ultimate demise. A noteworthy application of this principle of compatibility is seen in the system being developed by the National Library of Medicine.¹²

SYSTEM DESCRIPTION

The functional capabilities of the Virginia Tech Library System (VTLS) have been developed in two software releases, with the third release soon to appear. The initial release met the needs associated with circulation control and also provided rudimentary access to the catalog and serials holdings. The present release has benefited from the use of the MARC format, and allows vastly improved catalog access and control. Release III, the comprehensive library system now being developed, will draw together acquisitions, authority control, and serials control with the current capabilities.

VTLS Release I

The initial release of the system was developed in 1976 to meet needs generated by rapid library growth. Circulation transactions had been increasing at about 10 percent annually for the previous decade and were straining the manually maintained circulation files beyond acceptable limits. The main library* at Virginia Tech is organized in subject divisions—each essentially “owning” one floor of a 100,000-square-foot facility. A 100,000-square-foot addition to the library had been approved. Because Virginia Tech's library has only one card catalog, some means was necessary to distribute catalog information throughout a facility that

*Only two quite small branch libraries (architecture and geology) exist on campus. In addition there is a reserve collection located in the Washington, D.C., area that supports off-campus graduate programs in the areas of education, business administration, and computer science. All these sites are linked to the system.

was to double its size. After reviewing the alternative means of distributing the catalog—e.g., a duplicate card catalog, photographic reproduction of the catalog, or a COM catalog—it was decided to attack both problems, circulation control and remote catalog access, within a single online system.

VTLS was installed on a full-time basis in August 1976. Its first release ran continuously on the library's dedicated Hewlett/Packard 3000 minicomputer until December 1979. At that time the system held brief bibliographic data for approximately 325,000 monographs and 25,000 journals and other serial titles—records for about half the collection. While the first release ably met its goals, it became clear that it would prove to be an unsuitable host for additional modules involving acquisitions and serials control, primarily because of the brief, fixed-length bibliographic records. As a result of highly favorable price reductions in computer hardware and improvements in capability, it was possible to think in terms of storing one million MARC records online as well as supporting the additional terminals required for a comprehensive library system.

VTLS Release II

VTLS runs under a single online program for all real-time transactions. The major goals in the design of this program were the following:

1. Two conflicting requirements had to be accommodated: First, the program had to be easy to use for library patrons. This is requisite for a system that will eventually replace the card catalog. Second, the program had to be practical, efficient, and versatile for its professional users. The keystrokes required had to be minimal, and related screens had to be easily accessible from one to another.
2. The response time had to be good, especially for more frequent transactions.
3. The contents of all screens had to be balanced to provide enough information without being overcrowded and difficult to read or comprehend. Further, each screen of VTLS had to be arranged by some logical arrangement of the data it contains—for most screens this meant alphabetical sorting of the data according to ALA rules.
4. The format of all screens, especially those to be viewed by the patrons, had to be visually pleasing. Thus, the use of special symbols (which are so abundant on many computer system displays), non-standard abbreviations, and locally (and often quite arbitrarily) defined terms were unacceptable.
5. The program had to have security provisions to restrict certain classes of users from addressing particular modules of the program.

Considerable effort was spent to satisfy these goals. The first goal was achieved by the "network of screens" approach. The second goal—prompt system response—necessitated the use of the "data buffer

method," which, in turn, proved to have other uses (both of these techniques are discussed below). To satisfy goals three and four, a committee of librarians and analysts spent months drafting and reviewing each screen until it was finally approved by the design group. Goal five—security provisions—was reached without much difficulty.

Network of Screens

VTLS's data-access system is designed to be used as easily as a road map. This is accomplished by the use of a "network of screens." The network of screens is much like a road map in which a set of related data (a screen displayed in one or more pages) acts as a "city," and the commands that lead from one set to another act as "highways." VTLS has nineteen screens including various menu screens, bibliographic screens (see "The Data Buffer Method" below), serial holdings screens, item (physical piece) screens, and screens for patron-related data.

The user can "drive" from one "city" to another using system commands. The system commands are either "global" or "local." Global commands, as the name implies, may be entered at any point during the execution of the online program. A local command is peculiar to a given screen. Global commands are of two types: search commands and processing commands. Search commands are used to access the database by author, title, subject, added entries, call number, LC card number, ISBN, ISSN, patron name, etc. Processing commands, on the other hand, initiate procedures such as check-out, renewal, or check-in of items. The user first enters a global (search) command to access one of the screens in the network. From there, local commands that are specific to the current screen can be used. There are three different types of local commands: commands that take the user from one screen to another; commands that page within the current screen; and commands that update data related to the screen. For example, it is possible to start by entering an author search command to access the network and then proceed not only to find what books the author has in the system but also the availability of each of the books. If the books are checked out, information about the patrons who have them can also be reached. This display is called the patron screen. From the patron screen, one can "drive" to the patron activity screen, which displays circulation information about the patrons. Thus, each displayed screen leads to another. In fact, the searches can start at ten different screens and proceed in many different ways through the network.

Database Design

IMAGE/3000, Hewlett-Packard's database management system used by VTLS, is designed to be used with fixed-length records. This fact, coupled with the need to sort entries on most screens, created serious problems in the early stages of the system design. But various tech-

niques were devised to overcome these apparent road blocks.

Figure 1 illustrates the breakdown of the bibliographic record in the database and the way it is linked with piece-specific data. Bibliographic data are stored in three distinct groups for subsequent retrieval:

1. Controlled vocabulary terms. (Authority Data Set)
2. Title and title-like data. (Title Data Set)
3. All remaining bibliographic data; i.e., data that is not indexed. (MARC-Other Data Set)

This grouping of the MARC record extends to subfields, thus splitting mixed fields such as author-title added entries. When individual fields are parsed in this way, a single field may contribute more than one access point, such as variant forms of author, title, series name, subject, and added entries. Access by the standard bibliographic control numbers is effected by use of inverted files (not shown in the figure).

A fundamental characteristic of this layout involves the storage of controlled vocabulary terms (i.e., authors and subjects). Regardless of the number of references made to an authority term from different bibliographic records, the controlled vocabulary term is stored only once. The system assigns a unique number (Authority ID) to each such term and uses this number to keep records of the references made to it in a separate data set (Authority Bibliographic Linkage Data Set). This particular structure makes an authority control subsystem possible, speeds up online retrieval and display, and economizes mass storage.

The Data Buffer Method

The system displays bibliographic records in two different formats. If the terminal used is designated for librarians, the records are displayed

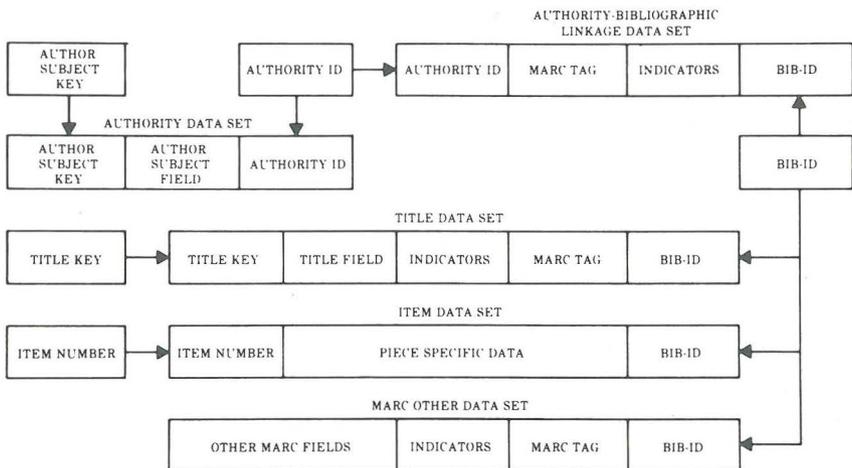


FIGURE 16. BIBLIOGRAPHIC LAYOUT OF THE CFS-II DATA BASE. (SIMPLIFIED)

Fig. 1. Bibliographic Layout of the VTLS Database (Simplified).

in the MARC format (the resulting screen is referred to as the MARC screen); otherwise, they are displayed in a screen that is formatted similar to a catalog card. Before displaying these screens, the online program collects and formats the data to be displayed and stores it in one of the two "buffer" data sets. The records stored in the buffer data sets are called buffer records. Buffer records can be edited, as required, by adding new lines, deleting, or modifying existing character strings. These updates can be executed quickly and without placing much load on the system since they involve little, if any, analysis, indexing, and sorting. Thus, the buffer data sets store all bibliographic updates and new data entry of the day. At night, these records are transferred to the rest of the database by a batch program.

The data buffer method has had several pronounced effects on the system. By transferring periods of heavy resource demand to off-hours, the system can work with full MARC records in a library that has a heavy real-time load of data entry, inquiry, and circulation. The data buffer approach also improves access efficiency because once a buffer record is prepared for a screen, subsequent searches for the same record are satisfied by the buffer record.

Data Entry and the OCLC Interface

The most frequently encountered method of entering MARC records into a local computer involves use of tape in the MARC II communications format. Alternative methods include the use of microprocessors or digital recorders which "play back" a MARC-tagged screen image from OCLC or some other bibliographic utility. These alternative methods have the strong advantage of shortening the delay introduced while waiting for a tape to be delivered.

We have been able to link the utility's terminal to the data buffer.¹³ Data flows from the utility to the buffer in real time. No intervention in the utility's terminal was required for the local processor to be able to capture the MARC-tagged screen. Batch programs running on the H/P 3000 read records from printer ports of OCLC terminals and pass them directly to the data buffer.

Once a record gets into the data buffer, it is accessible by OCLC number so that subsequent editing and linkage to piece-specific data or serial holdings can be made right away in the local system.

Buffer records can also be created by direct keyboarding of the full array of fixed and variable fields using the VTLS terminals.

Circulation

As with most other online circulation systems, VTLS uses machine-sensible bar-code labels to identify books and borrowers to the system. All efforts have been made to humanize the system. One consequence is

that the system does not make decisions better made by responsible staff. Thus, two kinds of circulation stations reside side by side. The first is staffed by students who typically work a ten-to-twenty-hour week and historically have shown high turnover. Their circulation stations only deal with inquiries and with heavily used but nondiscretionary transactions: check-out, renewal, and check-in. Should problems arise, the borrower is directed to the adjacent station staffed by a full-time employee who, using the system, can articulate circulation policy to borrowers and make decisions with regard to any questions concerning fines, lost books, or reinstatement of invalidated or blocked privileges.

START-UP

We found system start-up to be a relatively easy task. It was convenient to use the so-called rolling conversion in which items were labeled upon their initial circulation through the system. The greatest benefit was seen in the first year when the probability that items brought to the circulation desk were already known to the system increased exponentially. After six months this probability had risen to 65 percent with only 10 percent of the circulating collection having been labeled. At the end of the year the probability increased linearly at 0.7 percent per month. After three years of operation, the probability was 90 percent, with approximately 50 percent of the circulating collection having been labeled.

REFERENCE USE

The ability to distribute catalog access as well as circulation information provides a powerful information tool. A subset of all functions previously described is available to the nonlibrarian users of the system through user-cordial screens. A "help" function may also be initiated at any screen to guide users through the network of screens.

CURRENT DEVELOPMENT

Critical to the overall design of VTLS is the system's ability to treat serials and continuations. Without this capability, the modules being developed to support acquisitions, serials check-in and claiming, and binding, will not function satisfactorily. Equally important, the design lays the foundation for authority control by virtue of its use of a dictionary for all controlled vocabulary terms. Thus a name or subject entry is carried internally as a four-byte code, which is translated to the authority entry upon display.

Another internally coded data element, the BIB-ID, is designed to handle many of the linkage problems associated with serials and continuations. The BIB-ID is unique for each MARC record.

Prior to establishing the serials control modules governing receipt,

claiming, and binding, the coded holdings module must be functioning. This module will allow automatic identification of volume (or binding unit) closure and automatic identification of gaps in holdings or overdue receipts. Thus, highest priority has been given to the development of this module so that these other modules can, in turn, develop. The holdings module serves two functions: first, it allows the detailed recordings of serials holdings consistent with the principle stated earlier concerning microscopic data description; and second, these microscopic data are *coded* so that the system can recognize (and predict) particular pieces or binding units in terms of enumerative and chronological data.

The next three areas of development are modules for acquisitions and fund control, serials receipts and binding, and authority control. The final development will be comprehensive management reports.

It should be noted that each one of these developments will result in a specific benefit to the user community. The project is *incremental* in that the development of area *A* does *not* mean that area *B* must be developed for *A* to have lasting value. This incremental approach offers designers and administrators the advantages associated with an orderly growth in complexity and budget requirements. Further, the capabilities of the host hardware and software are stressed in smaller steps than would be the case if the comprehensive system were written and then turned on. The key move appears to be predefining the scope and capabilities of each stage so that a useful product emerges at its completion, and so that it lays a foundation for the next.

REFERENCES

1. Velma Veneziano and James S. Aagaard, "Cost Advantages of Total System Development," in *Proceedings of the 1976 Clinic on Library Applications of Data Processing* (Urbana, Ill.: University of Illinois Press, 1976), p.133-44.
2. Charles Payne and others, "The University of Chicago Data Management System," *Library Quarterly* 47:1-22 (Jan. 1977).
3. Audry N. Grosch, *Minicomputers in Libraries* (New York: Knowledge Industry Press, 1979), 142p.
4. Richard DeGennaro, "Wanted: A Mini-computer Serials System," *Library Journal* 102:878-79 (April 15, 1977).
5. John G. Christoffersson, "Automation at the University of Georgia Libraries," *Journal of Library Automation* 12:23-38 (March 1979).
6. Stephen M. Silberstein, "Standards in a National Bibliographic Network," *Journal of Library Automation* 10:142-53 (June 1977).
7. Network Technical Architecture Group, "Message Delivery System for the National Library and Information Service Network: General Requirements," in David C. Hartmann, ed., *Library of Congress Network Planning Paper*, no.4, 1978, 35p.
8. Arlene T. Dowell, *Cataloging with Copy* (Littleton, Colo.: Libraries Unlimited, 1976), 295p.
9. Michael Roughton, "OCLC Serials Records: Errors, Omissions, and Dependability," *Journal of Academic Librarianship* 5:316-21 (Jan. 1980).
10. Tamer Uluakar, "Needed: A National Standard for Machine-Interpretable Representation of Serial Holdings," *RTSD Newsletter* 6:34 (May/June 1981).

11. C.L. Systems, Inc., "The LIBS 100 System: A Technological Perspective," *CLSI Newsletter*, no.6 (Fall/Winter 1977).
12. Lister Hill National Center for Biomedical Communications, National Library of Medicine, "The Integrated Library System: Overview and Status" (LHC/CTB Internal Documentation, Bethesda, Md., October 1, 1979), 55p.
13. Francis J. Galligan to Pierce, 11 Feb. 1980.

Tamer Uluakar is manager of the Virginia Tech Library Automation Project. **Anton R. Pierce** is planning and research librarian at the university libraries. **Vinod Chachra** is director of computing resources and associate professor of industrial engineering.