Volume 4
Number 3 ISSN 0730-9295

Information Technology and Libraries

September 1985

CONTENTS

ITLBDC 4(3) 197–284 (1985); Inf Technol Libr

Information Technology and Libraries

Volume 4, Number 3: September 1985

CONTENTS

ITLBDC 4(3) 197–284 (1985); Inf Technol Libr

Information Technology and Libraries

Information Technology and Libraries is the official publication of the Library and Information Technology Association, a division of the American Library Association, 50 E. Huron St., Chicago, IL 60611; *Executive Director:* Donald P. Hammer. The journal is issued quarterly in March, June, September, and December.

Information Technology and Libraries publishes material related to all aspects of library and information technology. Some specific topics of interest are: Automated Bibliographic Control, AV Techniques, Communications Technology, Cable Systems, Computerized Information Processing, Data Management, Facsimile Applications, File Organization, Legal and Regulatory Matters, Library Networks, Storage and Retrieval Systems, Systems Analysis, and Video Technologies. *ITAL* welcomes unsolicited manuscripts. Submissions should follow the guidelines stated under "Instructions to Authors" on page 88 of the March 1985 issue.

Manuscripts of articles, communications, and news items should be addressed to: William Gray Potter, Editor, *Information Technology and Libraries*, Hayden Library, Arizona State University, Tempe, Arizona 85281. Copies of materials for review should be addressed to: Karin A. Trainer, *ITAL Book Reviews*, Yale University Library, 120 High St., P.O. Box 1603A, Yale Station, New Haven, CT 06520. Advertising arrangements should be made with William Z. Schenck, University of Oregon Library, Eugene, OR 97403.

Information Technology and Libraries is a perquisite of membership in the Library and Information Technology Association. Subscription price, $12.50, is included in membership dues. Nonmembers may subscribe for $25 per year. Single copies, $7.50.

Circulation and Production: American Library Association, 50 E. Huron St., Chicago, IL 60611. Please allow six weeks for change of address.

Publication of material in *Information Technology and Libraries* does not constitute official endorsement by the Library and Information Technology Association or the American Library Association.

Abstracted in *Computer & Information Systems, Computing Reviews, Information Science Abstracts, Library & Information Science Abstracts, Referativnyi Zhurnal, Nauchnaya i Tekhnicheskaya Informatsiya, Otdyelnyi Vypusk*, and *Science Abstracts Publications.* Indexed in *Computer Contents, Computer Literature Index, Current Contents, Current Index to Journals in Education, Education, Library Literature, Magazine Index, and NewSearch.* Microfilm copies available to subscribers from University Microfilms, Ann Arbor, Michigan.

Second-class postage paid at Chicago, Illinois, and at additional mailing offices. *Postmaster: Send address changes to* Information Technology and Libraries, *50 E. Huron St., Chicago, IL 60611.*

Electrons, Electronic Publishing, and Electronic Display

Edwin B. Brownrigg and Clifford A. Lynch

This article provides a perspective on electronic publishing by distinguishing between what we call "Newtonian" publishing and what we call "quantum-mechanical" publishing. This distinction revolves around the means by which information is distributed. We conclude that much of what is currently called electronic publishing is actually classical Gutenberg-style publishing carried out by modern methods and that, from our perspective, electronic publishing has been a reality for many years. Computers and recent advances in telecommunications, however, have greatly increased the capabilities of electronic publishing and have also given rise to the possibility of creating new types of publishable artifacts, some of which we describe. The article ends with an examination of some of the recent copyright-law issues and their relationship to quantum-mechanical publishing; in particular, this section explores the distinction between the replication and the display of a copyrightable work.

THE PROBLEM OF ELECTRONIC PUBLICATION

Suddenly people are speaking of the "advent of the electronic publishing age." Yet publishing has existed since the time of Gutenberg and, for nearly a century now, information has been disseminated electronically. One can argue that electronic publishing was established, at least in America, in East Pittsburgh in 1919–1920 with Frank Conrad's music broadcasts at radio station KDKA (previously station 8XK).[1]

In this paper we will explore the meaning of electronic publishing. We will argue that much of what today is called electronic publishing is firmly rooted in the printing press tradition, albeit updated to use electronic paper. Our thesis is that electronic publication is a delivery medium; publication is an action, a process, rather than an artifact. From the verb "to publish" comes the noun "publication," which means simply anything that is published. Calling something a publication says little about the nature of the thing, except that it can be reproduced or displayed at a distance. An electronic publication, in the strict sense, is anything that is published through electronic means; it says almost nothing about the nature of the work—the content of the media.

Our first task must be to distinguish the new electronic delivery mechanisms from the classical, Gutenberg, delivery mechanisms. We will then review some of the works that can be delivered through these two kinds of media. We will discuss not only updated forms of old printed works reworked for electronic delivery, but also new classes of works based on electronic technologies. The latter are not, from our point of view, merely electronic publications, but rather new kinds of works that can be delivered only over the channels of

Edwin B. Brownrigg is director of the Division of Library Automation, University of California, Berkeley, and Clifford A. Lynch is manager of the Computer Resources Group at the same division.

electronic publication. They are artifacts that exploit the capabilities of the electronic channel in essential ways. Finally, we will explore some of the ramifications of electronic publication for intellectual property rights and the copyright laws used to protect such rights.

MEDIA AND PUBLISHING

Since Gutenburg, publishing has primarily involved hard copy media, of which paper has been the most popular. Traditionally, each step of the publishing process—authoring, submission, referencing, editing, typesetting, printing, disseminating, storing, and delivering—has used paper.

How do paper and electronic media differ? To label the distinction, one might say that hard copy publishing is "Newtonian," and electronic publishing is "quantum-mechanical."

The difference between the two becomes apparent if we construct a simple thought experiment about document delivery over a relatively large distance. It is not relevant whether the publication was created or stored electronically. What is relevant is the method of delivery.

In this experiment we wish to deliver a published journal article to a space station in orbit 800 kilometers above the earth. The article could be stored either as hard copy (on paper) or digitally (on magnetic or optical media). Either form could be delivered from earth to the space station using the Newtonian or the quantum-mechanical method.

With the Newtonian method, either the hard copy or the digital medium could be put in a pouch and sent via rocket. The minimal energy required to deliver the document is directly proportional to one-half the square of escape velocity (48,000,000 centimeters per second) from earth's gravity, or about 24,000,000,000,000 ergs. In reality, we should multiply that relatively large number by one hundred to account for the inherent inefficiencies of the rocket.

With the quantum-mechanical method, the information content, without the storage medium, would be sent via radio wave from earth to the space station (and perhaps re-stored there; we shall return to the question of re-storage later). If the original storage media had been paper, the content would be digitized and modulated onto the radio carrier wave. If the original storage media had been magnetic or optical, the digitizing step of the process would be bypassed. If we assume that the radio transmitter on earth is 25 watts, the energy required to send the document would be on the order of 250,000,000 ergs. We should also multiply this relatively small number by one hundred to account for powering the radio receiver on the spaceship.

Thus quantum-mechanical delivery requires five orders of magnitude less energy than does Newtonian delivery. Moreover, quantum-mechanical delivery proceeds at roughly the speed of light and therefore outpaces the orbital mechanics of Newtonian delivery by several orders of magnitude. In either case the cost of building the delivery device (the rocket or the radio sets) has been left out of the calculation. Each cost represents capital investment, which varies according to other factors. However, radios are obviously far less expensive to build and operate than rocket ships.

Although Newton experimented with optics (for example, the diffraction of light through a prism), he did not treat light waves (photons) in his seventeenth-century physics. It was not until the late nineteenth century that Heinrick Hertz, James Clerk Maxwell, and others began to explore the possibility that light and radio waves were the same thing. Then in the twentieth century scientists began to understand the quantum-mechanical nature of radio waves (photons) and how they are created, modulated, transmitted, and detected. This understanding set the stage for the birth of electronic publishing. It is interesting to note that all attempts to understand electromagnetic effects within the confines of Newtonian physics revolved around the problem of "action at a distance." To resolve this difficulty, scientists developed various models of electronic aether—the luminiferous medium.[2] While the details of the physics have been greatly refined, the underlying idea of the early architects of the aether theories is still a valuable guide to understanding the possibilities quantum-mechanical media offer publishing.

Light and radio waves are photons. Indeed, the photon is classified as a stable, long-range field particle with a mass of zero, no charge, and a spin of one. The electron, on the other hand, is classified as a stable lepton of the fermion variety, with the lightest mass of particles that have mass, with either a positive or negative charge, and with a spin of one-half. But the two particles work together in the universe as enantiomorphs.

Alternating currents of electrons are the source of one of the most vital entities of the whole of physics—namely, the electromagnetic wave. Whenever an electric charge undergoes acceleration, an electromagnetic wave is produced. Metal aerials carrying oscillating electric currents radiate radio waves at the same frequency as the oscillating electric currents. All electromagnetic waves travel in a vacuum at the same phenomenally high speed (3,000,000,000,000 centimeters per second). All such waves can induce a nonuniform motion of electric charge. Thus, oscillating electrons induce photons, which in turn induce oscillating electrons at some distance. This phenomenon, plus the ability of electromagnetic waves to propagate faster than anything else through nothing at all, makes them invaluable in communication. Waves are the mechanism of quantum-mechanical publishing. The control of electromagnetic waves, rather than the computer, made electronic publishing possible.

Computers and modern telecommunication technologies, however, have greatly enlarged the possibilities for electronic publishing. Prior to the introduction of these technologies, electronic publishing was primarily an ephemeral, uncontrolled process. Few people could request radio or television programs. Equally few could record these programs for later reuse; tape recorders and particularly video recorders are relatively new developments as consumer electronic devices.

In the late 1960s and 1970s computers and advanced telecommunications introduced several new opportunities. First, and most importantly, they opened up the possibility for point-to-point, two-way, electronic publishing—users could request and receive works through electronic publication on demand. Moreover, it became practical to store a copy of a work rather than just to see or hear it as it "passed by." This was the real start of publication on demand.

WORKS DELIVERED THROUGH ELECTRONIC MEDIA

Having described the medium of electronic publishing, we now examine the works that can pass across this medium.

The first works were radio broadcasts of audio signals. The printed word can be readily translated into speech and thus to radio; the full range of printed media can be easily translated into television images. Thus, by the time color television came into widespread use in the 1960s, electronic publishing could deliver all of the Gutenberg-era works—the printed word, pictures, photographs, and films—but there were few new, inherently non-Gutenberg works available.

It is difficult to characterize a totally non-Gutenberg work. In virtually all cases it is possible to deliver a work through Newtonian means (such as shipping a disk) and then translate it back to electronic form at the end of its journey (by putting the disk into a disk drive attached to a computer). A totally non-Gutenberg work would have to be changing constantly (so that any copy delivered through Newtonian means would be out of date); have "real-time" value; and/or involve "real-time" interaction between multiple participants. Multiuser, computer-mediated games would be examples of such works. Other examples might be community interactive videotext systems and LANDSAT images telemetered in real time.

There is a much larger and more easily identifiable class of work that can be delivered through either quantum-mechanical or Newtonian media but that requires the use of a computer. Today many such works are termed electronic publications. Computer databases, for example, use the electronic publishing channel and represent non-Gutenberg works. (Although they do not represent the same sort of intellectual property as do books, computer databases do represent major economic assets). Videotext systems—networks of complex, interactive color screens, sometimes with animation—also make intrinsic use of the electronic channel.

A particularly interesting form of a non-Gutenberg work is what could be described as the interactive computer book. For example, a program called SARGON will play chess with the user on the Apple Macintosh. It will also teach chess; it will play a large number of classic chess openings, give hints about moves, work out various chess problems, and let the user switch sides. Playing chess games with computers is not new; it has been done for over two decades and at times has been viewed as a sort of philosopher's stone in the quest to develop artificial intelligence. SARGON is in many ways an analog of the old-fashioned, learn-to-play-chess book, but it offers much more. Another interesting example is the series of books on dynamics written by Abraham and Shaw;[3] these books are accompanied by a series of matching computer graphics programs that illustrate the material in the books. In both of these cases a paper book is still needed with the computer program, but there is no inherent difficulty in making the book readable as part of the computer program.

In spite of the possibilities, we constantly seem to be trying to return to the comfortable, familiar world of Newtonian delivery media by using physically distributed storage media, such as mass-produced optical disks of databases, CD ROMs, PROMs containing computer games, and the like. In such cases an information storage device and information content, rather than information content alone, is sent from place to place through channels such as the U.S. Postal Service (the ultimate Newtonian distribution method).

ELECTRONIC PUBLISHING AND THE PRINTED WORD

Using electronic delivery media for the printed word offers a great many advantages; it eliminates the need for transient storage and opens the possibility of obtaining material only when it is requested. It also raises the specter of economic and contractual chaos as libraries—the historical repositories of the printed word—become distribution points or switching stations in a quantum-mechanical world.[1]

Consider the advantages of electronically distributed journals, where individual articles can be obtained from some central repository (managed by the author or publisher) on demand. The case for converting library journal collections to such a basis has already been studied.[5]

The argument is that few people read every article in a given issue of a given journal. In addition, on the average, approximately 35 percent of subscriptions to journals are library subscriptions; however, for journals with very limited circulation the percentage of library subscriptions is much, much higher. The library also pays to process the journal volumes and to provide shelf space for the journal on an ongoing basis. Moreover, small-circulation journals cost more than large-circulation journals. The end result is that articles in journals that are never or rarely read are enormously expensive. Thus electronic media serve not only as a breeding ground and delivery mechanism for new types of works but also offer tremendous advantages as a channel for the delivery of older works.

INTELLECTUAL PROPERTY: THE DARK SIDE OF ELECTRONIC PUBLISHING

Let us return to our imaginary space station orbiting above the earth. A work has been transmitted to this station by radio. The space station inhabitants might display the work as it is received in much the same way one listens to the radio or watches television. They might store the signal for delayed review, or they might save the information in permanent storage, or perhaps even modify it and retransmit it elsewhere.

Regardless of how a work is used, there is a clear economic need to protect the author's rights to the work. This is a foundation of copyright law. Under the 1976 Copyright Act (17 USCA), an author obtains rights over reproduction, derivative works, distribution, performance, and display. With quantum-mechanical delivery, however, we enter an ambiguous world of fine distinctions between the display of a work and its reproduction.

Arthur R. Miller and Michael H. Davis have summarized the new law in their book, *Intellectual Property: Patents, Trademark and Copyright in a Nutshell:*

> The owner of a copy . . . might publicly display the work by showing . . . [it] to others or by making . . . [it] available to the public at one location and be immunized in doing so by section 109(b). But, in order to protect the author of copyrighted works from having their products unfairly exploited by developing communications technology, Congress has granted the copyright owner the exclusive right to display the copy in a way that would make it available in either multiple images or to persons outside the actual physical location of the copy. Thus, multiple showings of copyrighted images through computer screens, or transmissions to multiple computer terminals would constitute a public display right and that is not immunized by the more narrow privilege granted by section 109(b) limited strictly to "no more than one image at a time, to viewers present at the place where the copy is located."[6]

One of the greater ambiguities in copyright law is the "fair use" doctrine. The issue commonly arises over photocopying in libraries. Fair use involves balancing public interests against the rights of the creators, with the deciding factors being the nature, extent, and purpose of the use.

A discussion of the complexities of this part of the law is beyond the scope of this work, but suffice it to say that fair use of display will probably ultimately parallel fair use of photocopying. Unfortunately, the exact scope of such permissible copying is not defined in the statute, and since electronic copying is so much faster than manual photocopying, the problem will only be exacerbated in the future.

Section 108 of the 1976 Copyright Act authorized libraries to make copies of certain works, implicitly recognized the right of others to make photocopies in libraries having photocopy machines, and immunized libraries from any copyright liability if certain notices are posted.

The subject of photocopying provoked much discussion in Congress during debates on the new act. However, an "Agreement on Guidelines for Classroom Copying in Not-For-Profit Educational Institutions," which is characterized by emphasis on brevity and spontaneity, was reached. While the guidelines emphasized that fair use should not substitute for the purchase of materials that are otherwise available, it is doubtful whether current law will adequately protect the creators of works that will be accessed electronically; brevity and spontaneity have entirely different meanings in this situation.

De Sola Pool, in his book, *Technologies of Freedom*, presents a pessimistic view of the future of copyrighting in a world of electronic publishing:

> The recognition of a copyright and the practice of paying royalties emerged with the printing press. With the arrival of electronic reproduction these practices become unworkable. Electronic publishing is analogous not so much to the print shop of the Eighteenth Century as to word-of-mouth communication, to which copyright was never applied.[7]

MANAGING INTELLECTUAL PROPERTY IN A QUANTUM-MECHANICAL WORLD

We have seen that the current trends in electronic document delivery through telecommunications networks have considerable precedent in earlier broadcast "publication" of works. We have further seen some of the difficulties in adapting current mechanisms of copyright law to this complex new world of quantum-mechanical media. In concluding, we must put the entire question of intellectual property into perspective.

Ultimately, intellectual property legislation—patent, copyright, or trade secret—is simply a tool for dealing with an economic and social problem. As a matter of policy, it is desirable to promote the production of novel creative works. To promote such production, a mechanism must be established to compensate the authors for creating these works; the compensation should be linked to the "value" of the work as established by the marketplace.

Perhaps the management of copyright over display should be patterned after the management of certain performance rights rather than after the management of reproduction rights to printed material.

For the performance of musical compositions, the United States Congress in 1897 passed legislation that included public performance rights within the copyright statute. For many years ASCAP (the American Society of Composers, Authors, and Publishers), BMI (Broadcast Music, Inc.), and Sesac have been carrying out the essential clearing functions without which there would be endless searches and bargaining for performance rights, involving the owners of both established and obscure songs. Without these central licensing agencies, each radio station would require copyright clearance experts and would undergo programming delays while contacting the owners for each performance. Just as it became necessary to have general clearance agencies for performance rights to large amounts of musical material in order to facilitate the orderly supply of music to radio broadcast stations, it could be necessary to have such clearance agencies for display rights to large amounts of documents in order to facilitate the orderly supply of copyrighted material to database servers or libraries.

The way clearing agencies for display rights of copyrighted material would function could be patterned after the way clearing agencies for performance rights of copyrighted material function. Authors and publishers could submit their copyrighted material to the catalog of a clearing agency, which would collect royalties on their behalf according to the following scheme. First, the clearing agency would collect an assessed fee from database servers, libraries, and other agencies in the document delivery business. This fee would be based reasonably on some statistic, such as a small percent of gross receipts in the case of the database server, or ARL statistics in the case of the library. Next, the clearing agency would statistically monitor the electronic delivery of particular copyrighted material from individual businesses and institutions involved in document delivery. Certainly, with all transactions being carried out through computer, an abundance of statistical information on which to base payments should be available, although providing sufficient auditability to satisfy all interested parties is an extremely difficult problem. (Perhaps we can see at least a glimmer of a potential solution in various emerging technologies, such as write-once optical media and various authentication protocols, which are being developed using public-key cryptosystems as a basis.)[8]

Privacy is another issue. One of the great perils of electronic document delivery is that the identity of a document user is too easily recorded. The ability to defuse the privacy problem by paying royalties on a very broad statistical basis is very attractive. It could be done by using the number of times a document was accessed for display from a server without regard for the identities of the requesters. Finally, royalty payments to the copyright holders by the clearing agency could be based on a proportion of the amount of use of a particular copyright owner's material as against the total current use of the entire catalog of the clearing agency.

It is interesting to view the current Copyright Clearance Center (CCC) in the historical perspective of organizations such as ASCAP and BMI. There is a clear trend in the direction we are describing. The CCC began as primarily a passive accounting mechanism, but recently has begun to negotiate broad-based usage agreements with major corporations such as General Electric and Warner-Lambert.[9]

Whether such clearing agencies would become involved in granting advances of money to publishers and writers would depend on policy and practice. Nonetheless, the display of works through quantum-mechanical media does seem to have strong precedence in the history of performance of copyrighted material, and currently there is a legal and practical vacuum that must be filled. The distinctions between display, performance, and replication may ultimately prove to be of little use with electronic media and the new kinds of works that inhabit these media.

ACKNOWLEDGMENTS

We wish to thank Paige Cooke and Mary Jean Moore for editorial assistance.

REFERENCES

1. Laurence Bergreen, *Look Now, Pay Later: the Rise of Network Broadcasting* (New York: Mentor Bks., 1980).
2. Edmund Whittaker, *A History of the Theories of Aether and Electricity*, 2v. (New York: Harper, 1960).
3. Ralph Abraham and Christopher D. Shaw, *The Visual Mathematics Library*, V.1, *Dynamics—the Geometry of Behavior, Part 1: Periodic Behavior;* V.2, *Dynamics—the Geometry of Behavior, Part 2: Chaotic Behavior* (Santa Cruz, Calif.: Aerial Pr., 1983).
4. Edwin Brownrigg, Clifford Lynch, and Mary Engle, "Technical Services in the Age of Electronic Publishing," *Library Resources and Technical Services* 28:59–67 (Jan./Mar. 1984).
5. Donald W. King, *Electronic Publishing and Its Implications for Libraries*, ERIC Document (ED 196 435, October 4, 1979).
6. Arthur R. Miller and Michael H. Davis, *Intellectual Property: Patents, Trademarks and Copyright in a Nutshell* (St. Paul, Minn.: West Pub., 1983).
7. Ithiel de Sola Pool, *Technologies of Freedom*, (Cambridge, Mass.: Belknap, 1983).
8. Manuel Blum, "How to Exchange (Secret) Keys," *ACM Transactions on Computer Systems* 1:175–93 (May 1983).
9. *Advanced Technology & Libraries* 13, no.11:2 (Nov. 1984); 13, no.12:2 (Dec. 1984). ■■

The Systems Development Life Cycle as a Planning Methodology for Library Automation

David Cheatham

The systems development life cycle (SDLC) supports operational and managerial planning of automation projects. Its methodology, however, does not appear in library and information science literature. Notably absent are the techniques and tools of structured analysis and design. This neglect is unfortunate because the attention given to planning, particularly at the operational level, seems inadequate to ensure successful project development. The literature ignores the evolution of systems development methods that has occurred in business data processing during the past decade, and it fails to present a comprehensive, detailed methodology. Several corrective measures, including field reports, are needed.[1]

CHALLENGES OF AUTOMATION

Automation of library functions challenges administrators and staff throughout the project development process. Some challenges may even keep a project from getting started. Uncertainty as to need, cost, and impact on the total library system may keep libraries from investigating automation.[2,3,4] Other challenges threaten the success of projects that do get started. A vendor's or colleague's praise of a system may predispose a library administrator to automate, perhaps leading to a premature decision on which system to buy.[5,6] The complexity of bibliographic control and subtlety of bibliographic procedures may be underestimated and thus inadequately defined.[7,8] Questions without clear-cut answers may arise.[9] One such dilemma that many projects present is the choice between a proven but possibly obsolete system and one that is state-of-the-art but untested.[10] Decisions based on intuition may prove inadequate with time and experience. Once the system is installed and running, unanticipated outlays may be required for maintenance and upgrade.[11] After consuming substantial time, energy, and money, the new system may simply mechanize a former manual procedure without an improvement equal to its cost.[12]

Failure to successfully meet these challenges can be expensive in at least two ways. Automation in libraries often supports increased levels of service at lower costs than otherwise would be incurred.[13] Administrators who do not seriously consider automation may face unnecessarily high costs for improvements in service and thus, whether they decide to seek or forego the improvements, may lose the benefits of an automated solution. Those benefits also may elude administrators and staff who make a wrong project decision or overlook a decision that should be made. Experience has shown that mistakes often lead to escalating expense and diminishing performance.[14]

David Cheatham is a recent graduate of the Graduate School of Library and Information Science at the University of Washington.

SYSTEMS DEVELOPMENT LIFE CYCLE

In response to a similar set of challenges, analysts and engineers working primarily with business information systems have developed comprehensive planning methodologies. What began as systems analysis and design has been expanded in scope to a systems development life cycle (SDLC). This generic concept has taken several specific forms: I adhere to the structured methodology. The SDLC has two major benefits: it permits management to control the system development process with a realistic timetable and increasingly accurate cost estimates, and it permits the systems staff to create a complex system in manageable units.[15] Units are defined by proceeding in iterative stages, moving from general to specific levels when possible, and modularizing functions. Central to the SDLC are structured analysis and design. Structured analysis includes techniques and tools that should lead to an accurate and complete description of requirements for the new system. As DeMarco explains, its objective is "to cope with the most critical risk areas of analysis."[16] Procedures and tools of structured design maintain the integrity of system requirements while translating them into a model for implementation.

The SDLC addresses the challenges mentioned above. By limiting commitment of resources to the stage at hand and starting with a relatively inexpensive feasibility study, the efficacy of automation can be investigated. The context of each automation project is considered to be unique; solutions therefore are chosen only after thorough analysis. Graphic representations reveal areas of current and proposed systems inadequately understood, while iteration permits confusing questions to be left temporarily unanswered. Creation of abstract models of the old and new systems delays commitment to a specific physical solution. During this delay alternative solutions can be designed, with the result that mechanization need not be the default, an obsolete solution may be avoided, and performance should be optimized within the given constraints. Throughout the entire process, increasingly detailed approximations of the solution should permit correspondingly accurate cost estimates.

SDLC MISSING FROM LITERATURE

Despite the potential value of such a methodolology to the development of library automation projects, a review of journals and books shows that the SDLC has not been disseminated within the library and information science community. Some of its stages, methods, and tools have been described, but these seldom include the elements that distinguish SDLC from other planning methodologies, such as scientific management. Moreover, they often are presented without reference to any larger methodology whatsoever. This neglect is unfortunate because the published information on planning for library automation appears inadequate to ensure effective operational procedures.

This judgment may seem surprising because planning for library automation in general has been widely promoted. The need for planning is stated in reports on specific projects as well as in planning guides,[17,18] and it is recognized even when the solution likely will be a turnkey system.[19] The scope that has been defined for planning encompasses the entire project from needs analysis to system maintenance.[20,21] A long-range automation plan, with which individual projects should be in accord, is recommended.[22,23] Methods for planning such components as staff training and the user interface are explained.[24,25] Many sources describe planning tools, such as the critical path method, flow charts, and cost analysis.[26,27,28]

Yet such information has limited value as currently presented because neither individual contributions nor their sum represent a complete methodology, "an integrated set of tools and techniques."[29] At one extreme, the call for planning may appear as a cautionary note, almost an afterthought, within the description of an implemented system.[30] These warnings merely alert the reader to the need for planning. At another extreme, consideration of planning may involve a list of specific guidelines.[31,32] Insofar as they fail to consider the unique context of each individual library, these guidelines only target areas of concern. Between these extremes lie the summaries of an extended planning process and expositions of individual planning techniques.[33,34] The former provide an overview that must be supplemented with details to put the concepts into practice. The latter provide details but require an overview to determine where they fit. Unfortunately, there is as yet no link between the pieces. There is no comprehensive, step-by-step text, and the absence of references from articles and books to a methodology and related publications makes an integrated and complete set of methods difficult for the reader to construct.

Furthermore, the attention given to planning often concentrates on only part of the process. There has been an overwhelming concern that in planning for automation one must understand "the new electronic technologies, their problems, future and their impact on the library."[35] Special journal issues on information technology examine systems at general and specific levels.[36,37] Books aimed at encouraging effective planning survey hardware, software, and the impact of automation on libraries.[38,39] Reports on systems implemented in libraries expound upon features and functions.[40,41] This information on the technology itself provides a general context in which specific automation projects may be considered. It may also suggest alternative solutions to a problem faced by an individual library. But by emphasizing "what" instead of "how," such information implicitly disregards the unique character of each problem that might justify an automated solution. As in the case of one article promoted by the editors as a model of careful planning, it suggests that planning need only look outward to see what technology is available and what libraries are doing with it.[42]

EVOLUTION OF SDLC

Planning for automation must, of course, look inward as well, to clearly define organizational objectives, constraints, problems, data and flows of the current system, and requirements of a new system if there is to be one. This is the perspective of systems analysis and design, which emerged as an engineering tool in the early 1960s.[43] Almost immediately it was recommended as a desirable approach to automation projects in libraries. In 1962, before the methodology had congealed, leaders in the development of information systems conducted a workshop "not only [so] that the task and responsibility of the systems designer would be defined but, in addition, that some picture of the needs for education and research in the field of systems design would be made evident."[44] Reports reflected the formative stage of the methodology, suggesting motivations for its use and describing components, particularly the identification of user needs.[45,46] Schultz submitted the outline of a multiple-step system development process that clearly defines preliminary analysis (i.e., a feasibility study) and incorporates the concepts of iteration and discrete phases.[47]

From that early beginning, refinement of systems analysis and design for libraries and publication of its methods proceeded rapidly. By 1967 a few libraries, such as Rensselaer, had implemented a forms-driven approach.[48] In 1969 and 1970 books were published that explained a comprehensive methodology, detailed enough to permit readers to put it to use.[49,50] A few years later an incisive discussion of systems analysis and design comprised an integral part of introductions to library automation and data processing for libraries.[51,52] It was possible in 1973 to make the claim, at least, that systems analysis in libraries had become routine.[53]

Although the specific presentation varies from author to author, the process of systems analysis and design broadly encompasses the following stages designated by Hayes: "1. Feasibility analysis. 2. Requirements specification. 3. Detailed design. 4. Computer programming and check-out. 5. Computer program functional test. 6. Installation and implementation. 7. Maintenance."[54] It incorporates many standard planning tools such as Gantt charts, cost analysis, and interviews, but relies heavily on flow charts for analyzing the old system and designing the new one. A review of current descriptions confirms that the representation of this methodology as applied to libraries has not changed significantly in more than ten years.[55,56]

But as Robinson recognized early on, systems analysis and design is evolutionary in nature, with new techniques and tools added over time.[57] In the past decade the methodology has incorporated what are known as structured tools and techniques. The structured approach addresses an overriding problem that became clear only after some experience with automation: "for every dollar spent in true development (designing and coding), three dollars are spent in revision, either before or after delivery."[58] Based on the estimation that costs of correcting an error double as the project progresses to each successive phase, analysis of both the current and proposed systems takes on increased importance. Its tasks are expanded and redefined to decompose functions in terms of their data flows. Logical, or abstract, levels of analysis and design intercede between detailing the physical implementations of old and new systems. Graphic representations thus can be manipulated until an accurate, and, in the case of design, efficient model has been constructed. Data flow diagrams, decision tables or trees, and program structure charts or structured English largely replace flow charts as tools with which to clarify and illustrate the models. Specifications, even during design, are subject to modification as users and system staff review them. Structured analysis and design make up the heart of an extended development process, recently conceived of as the SDLC.

This is not the place to describe structured analysis and design in detail. For now librarians will have to refer to texts with a business orientation.[59,60] The point is that significant improvements have been made to the planning methodology and yet, as Parker observes, they rarely have been cited in the literature of library and information science.[61]

CORRECTIVE MEASURES

Acknowledgement of refinements to the system development process will provide a start, but will not prove sufficient to meet the challenges encountered with library automation, noted above. As King recommends, the methodology should not be considered ironclad but should be tailored to each project.[62] Library functions, files, and traditions have characteristics that appear to distinguish them from their counterparts in business data processing.[63,64,65] Files, for example, may be large with complex records, most of which are accessed infrequently. What is needed is customization of the SDLC to the library environment.

Such customization was, of course, a goal of the workshop on systems analysis conducted in 1962. The efforts to adapt the early methodology to library projects indicate much of what should be done now. First, we need a step-by-step text, oriented to practitioners and students of library systems, that illustrates the use of SDLC in a library. Second, descriptions of the methodology presented in books and articles on library automation must confirm its importance and represent its current state. Third, articles should thoroughly explain the latest additions and modifications to the techniques and tools as they become widely accepted by systems development staff in the business community, and their usefulness to library automation projects should be predicted.

There is a counterpart to the latter contribution that also seems necessary to successful automation planning for libraries. Weingard encourages increased publication of the field research performed continuously as librarians put theory into practice, or improvise, and observe the results of their activities. The description and evaluation of an implemented process would contribute to a fertile bed of shared knowledge based on experience.[66] A collection of final reports for systems projects conducted at the University of Michigan Libraries in the late 1960s serves as an early example of this type of contribution.[67] While current reports of library systems projects are published frequently, they rarely provide insight to the planning process followed at an operational level. Bolgiano and Parker offer two notable exceptions.[68,69] Each article serves as a case study, clearly illustrating how operational methods were applied.

Reports on the system development process practiced in the field can enrich the shared knowledge about planning for library automation at both specific and general levels. They can consider specific hypotheses that have been posed. Steen, for example, suggests that the additional expense incurred by writing technical as well as functional specifications for turnkey systems may prove cost-effective.[70] Conversely, Corey and Bellomy claim that shortcuts in the planning process can reduce development costs without endangering the success of many projects.[71] Field reports also may consider questions of a general nature. Minder states that the systems analyst in a library has broader responsibilities than the analyst in other environments,[72] while King indicates that an observable learning curve for mastery of the SDLC exists.[73] Evidence that confirms or disproves these hypotheses will be of great value.

CONCLUSION

Our effective use of technology historically has not kept up with its advances.[74] Such a lag should be no surprise with regard to automation of library functions, since librarians apparently have not mastered a process by which to discover and harness the power of the computer for their needs. De Gennaro predicts that computer-literate librarians and staff soon will have the knowledge to create innovative automated systems, but he fails to mention what that knowledge might be or how they will attain it.[75] The SDLC can provide the requisite knowledge of current systems and possible alternatives for successful operational planning. To use technology effectively, however, necessitates skilled managerial planning as well. We must intelligently ration our limited resources among possible projects and in the process establish a stable environment where change is measured and purposeful.[76,77] With its total systems view, SDLC offers management control of project development.

Much like automation, SDLC must be adapted to the requirements of individual situations. For small projects and those tightly constrained, an abbreviated version should prove sufficient. When the proposal is of a higher order, an integrated system or a new generation of computer applications, escalating challenges should be met with the extended methodology. The SDLC is an evolutionary set of procedures and tools that will continue to be refined. Library administrators and staff need exposure to the refinements as well as the basic planning process. The SDLC may represent a significant investment, but it should continue to pay off in satisfied users and staff, efficient operation, and lower maintenance costs long after the system is up and running.

REFERENCES AND NOTES

1. Three faculty members at the University of Washington contributed to this paper. William Burrows kindled my enthusiasm for systems analysis and design, while endowing me with a fundamental understanding of its concepts and techniques. Drs. Karen Sy and Edmond Mignon, along with Mr. Burrows, helped to focus several statements more clearly.
2. Patricia A. Armstrong, "Automated Circulation: for the Small Public Library," *Canadian Library Journal* 41:334 (Dec. 1984).
3. Donald J. Sager, "After the Planning Is Over: Implementing Computerization," *Show-Me Libraries* 34, no.3:13 (Dec. 1982).
4. Crystal Condous, "Cost-benefit Analysis and the Implementation of Automated Systems," *LASIE* 14:8 (Sept./Oct. 1983).
5. Ervin J. Gaines, "An Inquiry into Some of the Consequences of Database Sharing," *Library Journal* 109:448 (Mar. 1, 1984).
6. Bobbie Merilees, "RFPs and On-line Library System Selection," *Canadian Library Journal* 40:16 (Feb. 1983).
7. Allen B. Veaner, "Approaches to Library Automation," in *Reader in Library Systems Analysis* (Reader Series in Library and Information Science), eds. John Lubans, Jr. and Edward A. Chapman (Englewood, Colo.: Microcard Editions Bks., 1975), p.434; reprinted from *Law Library Journal* 64:146–53 (May 1971).
8. Susan Baerg Epstein, "Integrated Systems: Dream vs. Reality," *Library Journal* 109:1303 (July 1984).
9. Joel Robinson. "Automating the Library," *Show-Me Libraries* 35, no.10:20 (July 1984).
10. S. Michael Malinconico, "Planning for Obsolescence, "*Library Journal* 109:333 (Feb. 15, 1984).
11. Maurice J. Freedman, "Automation and the Future of Technical Services," *Library Journal* 109:1203 (June 15, 1984).
12. Cynthia I. Gozzi, "Local Systems: Design and Costs," *Journal of Library Automation* 13:169 (Sept. 1980).
13. Donald J. Sager, *Public Library Administrator's Planning Guide to Automation* (OCLC

Library, Information, and Computer Science Series) (Dublin, Ohio: OCLC Online Computer Library Center, 1983), p.17.

14. Robinson, "Automating the Library," p.20.
15. David King, *Current Practices in Software Development: a Guide to Successful Systems* (New York: Yourdon, 1984), p.2.
16. Tom DeMarco, *Structured Analysis and System Specification* (New York: Yourdon, 1978), p.9.
17. Judy Capes, "Computer Update—Another Retrospective Glance," *Canadian Library Journal* 40:299 (Oct. 1983).
18. Joseph R. Matthews, *Choosing an Automated Library System: a Planning Guide* (Chicago: American Library Assn., 1980).
19. Dennis Steen, "Computerizing Circulation Systems," *Ontario Library Review* 65:11 (Mar. 1981).
20. Rebecca Dunkle and Catherine McLoughlin, "Automating the Region: a Report on the Seminar," *Library Hi Tech* 2:43 (1984).
21. Susan Baerg Epstein, "Maintenance of Automated Library Systems," *Library Journal* 108:2312–13 (Dec. 15, 1983).
22. John Corbin, Developing Computer-based Library Systems (A Neal-Schuman Professional Book) (Phoenix, Ariz.: Oryx, 1981), p.24.
23. Malinconico, "Planning for Obsolescence."
24. Barbara Conroy, "The Human Element: Staff Development in the Electronic Library," *Drexel Library Quarterly* 17, no.4:91–106 (Oct. 1981).
25. John Vanderhoef and Paul Gallina, "Ergonomics of VDTs," *Canadian Library Journal* 40:285–87 (Oct. 1983).
26. Peter Brophy, "Critical Path Analysis: a Library Management Tool," *Program: Automated Library and Information Systems* 17:204–16 (Oct. 1983).
27. Robert H. Burger, "Caveat Lector, Caveat Descriptor, or, What You Think Flow Charts Say Is Often Not What They Should Have Meant," *Journal of Academic Librarianship* 6:260–63 (Nov. 1980).
28. Stephen H. Peters and Douglas J. Butler, "A Cost Model for Retrospective Conversion Alternatives," *Library Resources & Technical Services* 28:146–62 (Apr./June 1984).
29. King, *Current Practices in Software Development*, p.6.
30. Farooq A. Khalid, "Automation in a Special Library in Kuwait," *Information Technology and Libraries* 2:360 (Dec. 1983).
31. "Selecting a Micro for Office and Management Applications," *Library Systems Newsletter* 3:73–75 (Oct. 1983).
32. Robert F. Bocher, "Microcomputers: Guidelines for the Planning Process and Software Selection," *Wisconsin Library Bulletin* 78:59–61 (Summer 1983).
33. Richard W. Boss, *Automating Library Acquisitions: Issues and Outlook* (Professional Librarian Series) (White Plains, N.Y.: Knowledge Industry, 1982).
34. Paul E. Baldwin, "Planning the Implementation of an Integrated On-line Acquisitions System," *Canadian Library Journal* 41:31–36 (Feb. 1984).
35. Betty J. Turock, "Technology and the Post-industrial Society: the Academic Library in the 1980s and Beyond," *Catholic Library World* 55:302 (Feb. 1984).
36. *Library Journal*, Information Technology Issue 109, no.11 (June 15, 1984).
37. *Library Journal*, Information Technology Issue 110, no.2 (Feb.1, 1985).
38. Mark E. Rorvig, *Microcomputers and Libraries: a Guide to Technology, Products, and Applications* (Professional Librarian Series) (White Plains, N.Y.: Knowledge Industry, 1981).
39. Joseph R. Matthews, ed., *A Reader on Choosing an Automated Library System* (Chicago: American Library Assn., 1983).
40. Sue A. Dodd, "Building an On-line Bibliographic/MARC Resource Data Base for Machine-readable Data Files," *Journal of Library Automation* 12:6–21 (Mar. 1979).
41. Ralph M. Daehn, "Technology and Microcomputers for an Information Centre/Special Library," *Canadian Library Journal* 41:39–43 (Feb. 1984).
42. Tom Smorch, "Saga of an RFP for an Automated System," *American Libraries* 15:397–98 (June 1984).
43. Thomas Minder, "Application of Systems Analysis in Designing a New System," *Library Trends* 21:553 (Apr. 1973).
44. R.M. Hayes, "Introduction," in *Information Systems Workshop: the Designer's Responsibility and His Methodology*, ed. Charles P. Bourne (Washington, D.C.: Spartan Bks., [c. 1962]), p.4.
45. John A. Postley, "Behavioral Factors in Information Systems," in *Information Systems Workshop*, p.83–95.
46. Harold Borko, "Determining User Requirements for an Information Storage and Retrieval System: a Systems Approach," in *Information Systems Workshop*, p.37–60.
47. Claire K. Schultz, "A System for Systems Design," in *Information Systems Workshop*, p.121–27.
48. Paul L. St. Pierre, "Systems Study as Related to Library Operations: Analysis, Evaluation, and Design," in *Library Automation: a State of the Art Review*, ed. Stephen R. Salmon ([Chicago]: American Library Assn., 1969), p.13–29.

49. F. Robinson and others, *Systems Analysis in Libraries* (Sympleglades, No. 2) (Newcastle upon Tyne, England: Oriel Pr. 1969).
50. Edward A. Chapman, Paul L. St. Pierre, and John Lubans, Jr., *Library Systems Analysis Guidelines* (New York: Wiley, 1970).
51. Edward M. Heiliger and Paul B. Henderson, Jr., *Library Automation: Experience, Methodology, and Technology of the Library as an Information System* (McGraw-Hill Series in Library Education) (New York: McGraw-Hill, 1971), p.105–58.
52. Robert M. Hayes and Joseph Becker, *Handbook of Data Processing for Libraries*, 2nd ed. (A Wiley-Becker & Hayes Series Book, Information Sciences Series) (Los Angeles: Melville, 1974), p.123–213.
53. Ruth C. Carter, "Systems Analysis as a Prelude to Library Automation," *Library Trends* 21:518 (Apr. 1973).
54. Hayes and Becker, *Handbook of Data Processing*, p.131.
55. Philip Schwarz, "Selecting Automated Minicomputer Turnkey Systems," *Wisconsin Library Bulletin* 78:70–76 (Summer 1983).
56. James Rice, *Introduction to Library Automation* (Library Science Text Series) (Littleton, Colo.: Libraries Unlimited, 1984), p.95–144.
57. F. Robinson, *Systems Analysis in Libraries*, p.11.
58. DeMarco, *Structured Analysis and System Specification*, p.298.
59. James A. Senn, *Analysis and Design of Information Systems* (New York: McGraw-Hill, 1984).
60. William S. Davis, *Systems Analysis and Design: a Structured Approach* (Reading, Mass.: Addison-Wesley, 1983).
61. Doreen V. Parker, "Logical Data Analysis for Library Systems Design," *LASIE* 14:9 (Nov./Dec. 1983).
62. King, *Current Practices in Software Development*, p.182.
63. Hayes and Becker, *Handbook of Data Processing*, p.126–27.
64. John J. Eyre, "Uses and Limitations of Computers in Libraries," in J. Stephen Parker, ed., *Aspects of Library Development Planning* (London: Mansell, 1983), p.94.
65. John H. Ashford and Derek Matkin, *Studies in the Application of Free Text Package Systems; for Information Storage and Retrieval in Libraries and Related Information Centres* (Case Studies in Library Automation) (London: Library Assn., 1982), p.3–4.
66. Darlene E. Weingand, "Editorial," *Library and Information Science Research* 5:237–38 (Fall 1983).
67. Barton R. Burkhalter, ed., *Case Studies in Systems Analysis in a University Library* (Metuchen, N.J.: Scarecrow, 1968).
68. Christina Bolgiano, "A Systems Approach to Label Production through the OCLC System," *Journal of Library Automation* 12:383–86 (Dec. 1979).
69. Parker, "Logical Data Analysis."
70. Steen, "Computerizing Circulation Systems," p.11.
71. James F. Corey and Fred L. Bellomy, "Determining Requirements for a New System," *Library Trends* 21:550–51 (Apr. 1973).
72. Minder, "Application of Systems Analysis," p.558.
73. King, *Current Practices in Software Development*, p.41.
74. Thomas J. Galvin, "The Significance of Information Science for the Theory and Practice of Librarianship," *Libri* 34:85 (June 1984).
75. Richard De Gennaro, "Integrated Online Library Systems: Perspectives, Perceptions, & Practicalities," *Library Journal* 110:40 (Feb. 1, 1985).
76. Richard De Gennaro, "Matching Commitments to Needs and Resources: Reflections on Managing the Academic Library in Hard Times," in *Library Lectures, the University of Tennessee, Knoxville* (Knoxville: Univ. of Tennessee, 1981), p.11.
77. Kenneth Dowlin, "I Am Not Willing to Destroy My Library in Order to Change It, "*Library Association Record* 85:449 (Dec. 1983). ■■

The Automation of Reserve Processing

James Self

A number of automated library systems do not include reserve modules, although automation of reserve processing is technically rather simple and yields a large return in increased production and efficiency. This paper describes an automated reserve processing system developed locally at the Clemons Library, University of Virginia.

Reserve is an activity found in virtually every academic library. Numerous items have to be processed each term, and the requests to place items on reserve tend to come all at once—at the beginning of each term—creating backlogs, disorder, and a reduction in service. Many of the items placed on reserve are never or rarely used,[1,2] creating frustration and leading some to question the usefulness of the whole idea of reserve,[3] as it detracts from more appropriate library services.

Whatever the feelings of librarians, reserve seems here to stay. It is used by a significant number of faculty and students, who obviously feel it is an important service. Thus it would seem worthwhile to devise a reserve system that works as efficiently as possible.

Reserve is a labor-intensive activity of high visibility. Speed of processing is often important; the books need to get on reserve fast. It would seem an ideal candidate for automation. Some automated circulation systems include reserve modules, but a number of systems do not yet offer a reserve feature. They may have a very elegant means of circulating ordinary books, but the repetitious, labor-intensive reserve activity is still carried out manually.

Why are there circulation systems in use without a reserve component? One reason may be the low esteem of reserve; librarians keep hoping it will go away. But another reason becomes apparent when we examine the reserve process. It is not one process, but two distinct activities:

1. Processing: preparing records so that patrons will know what items are on reserve and how to find them; in a traditional manual system, several cards are typed for each item. One card is filed by call number (shelf list), one by professor or course number, and one by author or title.

2. Circulation: checking out books; in a manual system, this is accomplished by having a patron fill out a slip or sign a book card for each transaction. The cards are then filed by call number.

The first of these activities, processing, is relatively easy to automate. Rao and Jones describe a batch system, using punch cards, introduced at Eastern Illinois University in 1973.[4] Earlier articles by Fasana,[5] Gallivan,[6] and Simmons[7] also described a reserve processing system that replaced the typing and filing of multiple cards with computer printouts. In all these systems, savings in labor and reductions in turnaround time resulted. Such a program, as described by Rao and Jones, is relatively simple, but once the initial programming is done, it is effective.

The other operation in reserve—

James Self is director, Clemons Library, University of Virginia, Charlottesville.

circulation of materials for short periods of time—is technically a much greater challenge to automate, especially when done as part of an automated circulation system. In an automated circulation system, books on reserve are checked out to "Reserve" or to the "Reserve Room." The problem is how to check out the item to a patron for two hours or two days without checking it in from reserve. Some of the circulation systems have solved this problem and offer an integrated reserve component, but in other systems the technical problem has not been solved and reserve has to be done manually.

The designers of circulation systems have apparently viewed reserve as a single, or integrated function. Thus automated systems do not have a reserve processing function unless they also have a reserve checkout function. The processing function is technically simple, with a large payoff in service and efficiency, but this function is not offered until the technical challenges of the reserve checkout function are overcome. Our experience at the University of Virginia indicates that libraries and systems designers would do well to consider reserve processing and reserve circulation as distinct operations.

Clemons Library opened in March 1982 as the undergraduate, high-use library at the University of Virginia. An online circulation system developed by the university's Administrative Computing Center (ADCOM) was in place. It used an IBM 4341 mainframe and IBM 3270 terminals, with OCR readers, book labels, and borrower cards. A log of each transaction is kept on disk or tape, and the circulation records can be read by EASYTRIEVE or Statistical Analysis System (SAS).

The circulation system was very efficient for ordinary checkouts, but it did not include a reserve component because ADCOM was frustrated by the technical challenges of the reserve checkouts and, also, because the university had determined other priorities for ADCOM. Thus, a reserve component was not developed in 1982.

Prior to the opening of Clemons Library, the main reserve operation was located in Alderman Library, the university's main library. When Clemons opened, the manual reserve system was moved there from Alderman.

In August and September 1982, the transferred system was put to a severe test. Because of staff reallocations caused by the establishment of an entirely new library (Clemons), the reserve system was operating with one less staff member. Also, searching and fetching now took place in two buildings rather than one, adding to the workload. When the faculty began submitting large numbers of reserve requests for the fall semester, the manual processing system simply failed. Large backlogs quickly developed, and because of the complexity of the manual system, only a limited number of staff members could productively work on processing the materials. It is unnecessary to enumerate the gruesome details, but it was obvious that a new processing system was desperately needed. We set a goal of having an effective automated processing system in place by fall 1983.

We considered developing a system using a microprocessor but soon realized there should be some way to utilize the existing circulation system. The books checked out to reserve had records in the circulation system that included author, title, call number, and a unique identifier for each item (the item number). Using SAS to read the transaction logs, it was a very easy matter to get separate printouts of all books on reserve, sorted by author, title, and call number. No data entry was required to do this. With such printouts available, we questioned whether we needed to continue typing and filing cards in the reserve author catalog and the reserve shelflist, especially since typing and filing backlogs were a primary source of problems in fall 1982. Before we could stop typing cards, though, we had to consider what to do about photocopies. There were many photocopied articles on reserve, and these articles were not on the circulation database. To get them on the reserve printouts would require us to add a record for each article to the circulation database. The record would need to include author, title, an item number, and a dummy call number that would not conflict with any call number already in the database. After a little testing, we were able to say with confidence that entering a dummy record for a photocopy would certainly take no longer than typing and filing a set of cards for the same item. Thus we could produce author, title, and shelflist printouts for all items on reserve and eliminate one of the major causes (typing and filing) of the backlog.

We were left with one problem. At most libraries, including ours, patrons have traditionally had access to reserve materials through the name of the instructor and the course number. Nothing like that was on the database, so we began to look for a place to store such information. Each item record includes a holding library symbol, a loan period designator, a material type designator, and a space for a free-form note. We were prepared to use the note field to hold the reserve information, although there would have been some complications. It would have necessitated beginning each reserve note with a delimiter (perhaps a "¢" sign) to distinguish reserve notes from other item notes. It would also have caused some difficulty in fixing a standard format in a free-form field.

As it turned out, we did not have to solve these problems because ADCOM agreed to add another field, a reserve note, to the item record. The programming required to establish a new field was minimal and did not require any shifting of priorities within ADCOM. So, a reserve note was created with a fixed format of fifteen characters for the surname of the instructor, eight characters for the course number, two characters for the loan period (2H or 2D), and four characters each for the beginning semester and the ending semester (e.g., FA83).

The programmer at ADCOM had a very helpful observation at this point. He suggested that we limit use of the reserve note to items checked out to reserve. If someone had forgotten to check the item out to reserve, the system, by not accepting the note, would indicate that something was wrong. The same thing would happen if an item number was miskeyed; the system would not accept the reserve note. ADCOM agreed to make one exception to this rule; reserve notes could be added to items recalled for reserve.

With the addition of this function, we now had all the information we needed to produce the printouts (or catalogs) needed in a reserve operation—listings by call number, author, title, and instructor/course. We also had the means for keeping a history of all books placed on reserve.

All of this had been done by early spring in 1983, so we revised our procedures and work flow to implement the system in summer 1983. The reserve procedures were completely redone to make maximum use of the automated processing. Also, the new procedures were very segmented and specific. An employee could be trained in a short time to perform a specific function without having to learn any other function. Thus, if a backlog developed in one area, we could quickly shift staff into that area until the backlog disappeared. Because the summer session at Virginia attracts few students, the real test of the new system came in fall 1983.

The results were very gratifying. In fall 1982 turnaround times were measured in weeks; in 1983 it was a matter of days. Even during the rush of the first week of classes in 1983, there was no backlog. All incoming requests were handled within one or two days; over 90 percent of all requests were filled and on the printouts within three days of receipt of the request. Those that took longer were generally recalls and missing books. The reserve backlog was only a memory.

The dramatic change in performance of the reserve operation was due to automation of the processing function. The labor of typing and filing reserve catalog cards was saved. Most of the data formerly typed onto the cards was sitting in the computer ready to be printed out. The data that had to be added, the course/instructor information, and the bibliographic information for photocopies was keyed into the terminal much faster than cards could be typed.

The new system functions in many ways like those described by Rao and Jones, but it requires less labor than the earlier systems because the author, title, and call number of each book do not have to be keyed in. That information is already in the circulation system.

With the success of the new system in the Clemons Library, the same system was implemented in the Fine Arts Library, the Science Library, and the Education Library in 1984. As other branch libraries are brought onto the circulation system, it is expected they will implement the reserve system as well.

The system produces a variety of printed reports. ADCOM automatically produces weekly printouts by author, instructor/course, and call number. Each week we request additional runs and copies as needed. During the first month of each semester, when items are constantly being added to reserve, we receive daily a call number printout, an author printout, and two copies of an instructor/course printout. We divide the two copies of the instructor/course printouts into two parts (A–L and M–Z), so that four patrons can use that printout at once. Once a week, using an in-house SAS program, we also produce a printout by title.

Using SAS, we have written programs inhouse that enable us to identify problems and errors. Once a week we produce a list of all items recalled for reserve more than ten days ago. We then prod the borrower again, find another copy of the item, or notify the instructor of the problem. We also receive a list of all items on current reserve that were taken off reserve (checked in) during the past week. If there is no note explaining why the item was taken off reserve, we put it back on. Another report lists all books checked out to reserve during the last week that do not have the instructor/course information. Items on the list are investigated, and the proper information is added. To give us an idea of who is using the reserve system, we produce a list of instructors and courses and the number of items on reserve for each course. This list enables us to see who is "overusing" the system. It also lists separately the courses with only one item on reserve. We investigate each of these; quite often, a keying or a spelling error has occurred.

We are able to print out at any time a list of items on reserve for any particular instructor, course, and semester. Each semester we send lists of last year's reserve items to instructors, who update them by adding or deleting items. We use the lists as worksheets when placing items on reserve for that instructor. The lists indicate which items are still on the reserve shelf and which need to be fetched from the stacks. Items in the first category do not have to be touched; we merely key in the instructor, course, loan period, and semester.

Using the history tape and the current reserve notes, each semester we produce a "purge list" of items, in call-number order, to be removed from reserve. To remove the items from reserve, they are merely checked in on the circulation terminal. The program used to generate the list is flexible; if we wish, we can remove all items not currently on reserve, or we can allow a time lag for certain items or for items from specific libraries. Using a purge list to remove items is of course much more efficient than using a shelflist printout. To use a shelflist, one has to look at each record and determine if the item should be removed. When a purge list is used, the computer has already determined what is to be removed; the library staff member sees only the call numbers of items that are ready to be taken off reserve. There seems to be no limit to the variety and number of reports available through SAS.

As noted above, one of the reasons for the problems in 1982 was a reduction of 1 FTE in the reserve staff. After using the new system for a year, we have been able to reduce the reserve staff by another .5 FTE. Thus, we are operating with a staff that is 38 percent smaller than in preautomation days, and turnaround time has been reduced dramatically.

During the first year of the new processing system's operation, the circulation of reserve materials was done manually. Each patron signed a card when an item was taken out. A reserve checkout system using a distributed processor, described by Buxton,[8] is now being implemented. The checkout system will enable us to add circulation data to our reports, and it will of course simplify and improve our service at the reserve desk. But, it is important to note that the automated processing system will work with an automated or manual reserve checkout system.

SUMMARY

Reserve automation consists of two distinct parts: automation of processing and automation of checkouts. From a technical point of view, the first is relatively easy to accomplish; the second is rather complicated and challenging. A number of circulation systems do not include a reserve module, apparently because of the difficulty of automating the reserve checkout function. But one can automate the processing function without automating reserve checkout. At the University of Virginia, we found that automating reserve processing produced a very high payoff in terms of reduced labor and improved service.

REFERENCES

1. Bob Carmack and Trudi Loeber, "The Library Reserve System—Another Look," *College and Research Libraries* 32:105–9 (Mar. 1971).
2. Larry Hardesty, *Use of the Library Reserve Collection at DePauw University* (Bethesda, Md.: ERIC Document Reproduction Service, ED 192 734, July 1980).
3. Norman D. Stevens, "On Our Minds; A Hard Look at Reserve," (New England Academic Librarians Writing Seminar) *Journal of Academic Librarianship* 4:86–87 (May 1978).
4. Paladugu V. Rao and Carole B. Jones, "System to Manage Reserve Texts (SMART)," *Journal of Library Automation* 9:328–34 (Dec. 1976).
5. Paul J. Fasana and others, *A Computer Based System for Reserve Activities in a University Library* (Bethesda, Md.: ERIC Document Reproduction Service, ED 035 431, Sept. 1969).
6. B. Gallivan and others, *Computer Listing of a Reserve Collection* (Lancaster University Library Occasional Papers, no. 6) (Bethesda, Md.: ERIC Document Reproduction Service, ED 104 335, 1972).
7. Peter Simmons, "Reserve Collections: Some Computer Assistance for the Perennial Problems," *Canadian Library Journal* 29: 82–87 (Mar./Apr. 1972).
8. David T. Buxton, "The Clemons Library Reserve Circulation System: A Model for Distributed Processing in Library Networks," *Information Technology and Libraries* 3: 178–82 (June 1984).

■■

Statistical Collection Management Analysis of OCLC-MARC Tape Records

Evelyn Payson and Barbara Moore

Using OCLC-MARC tape records, master records were created for collection management reports at the University of Wisconsin–Whitewater. These fixed-length records contained only certain fields from the OCLC-MARC records. Reports were generated to help evaluate collection growth by type of material, location, and subject, based upon the LC call number. In addition, SAS, a statistical package, was run against these records to generate information for special purposes, such as determining how the age of material affected circulation and evaluating the number of books purchased by given departments in certain subject areas.

INTRODUCTION

The University of Wisconsin (UW)–Whitewater has devised computer programs to extract and abbreviate specific fields from its OCLC tapes and create shorter, fixed-length records that can be used for producing a variety of collection management reports. Some of these reports are produced by the tape extraction programs themselves and others are available on user request, either through the extraction programs or through the use of a statistical package such as SAS. The information provided from the tapes can either stand alone or be used to supplement the statistics-generating abilities of an online system.

BACKGROUND

UW-Whitewater is located in the southeast corner of the state. It has an enrollment of 9,441 undergraduate students and 1,296 graduate students. The Department of Library and Learning Resources has over 350,000 volumes in its collection. The library began using OCLC in 1976 to produce catalog cards and, through an extensive retrospective conversion project, now has over 225,000 titles on OCLC archival tapes. The state university libraries in Wisconsin are currently working together to plan for and install online circulation and public access catalog systems.

At UW–Whitewater the responsibility for ordering library materials is divided between librarians and faculty. Librarians are primarily responsible for purchasing periodicals, continuations, reference materials, government documents, and some of the educational and media materials. Faculty order materials in their respective disciplines. A percentage of the library collection development funds is divided among university departments for use in buying library materials. Accounts of departmental collection development funds enable librarians to monitor the rate at which funds are being spent. No attempt is made to monitor how well departmental funds are used by ascertaining the quality of the materials purchased.

Evelyn Payson is director of library automation at University of Wisconsin, Whitewater; Barbara Moore is library systems analyst at Rochester Public Library, New York.

While this approach to collection development allows faculty to purchase the appropriate library material in their subject of expertise, it may also result in unbalanced collection growth for the library as a whole. Faculty may buy in narrower ranges than their disciplines cover; general subject areas that do not fall into one teaching department may be ignored; and some subject areas may have costlier materials, resulting in fewer titles being added to the subject.

To help the librarians evaluate how the collection had been growing, especially by departmental orders, a series of programs was written to analyze OCLC archival tapes. OCLC tapes contain bibliographic information concerning recent UW–Whitewater acquisitions in machine-readable form. UW–Whitewater receives an OCLC archival tape every three months. While the primary reason for obtaining the tapes is to create a database for a future, local, online circulation system and public access catalog, computer programs have been written to analyze the tapes in a batch mode to provide timely collection growth statistics.

LITERATURE REVIEW

A literature search revealed two articles on the use of OCLC tapes for collection analysis. Evans, Gifford, and Franz reported on a collection development analysis of OCLC archival tapes developed for the State University of New York's (SUNY) Central Administration Office of Library Services.[1] SUNY developed a series of programs to produce reports by LC, Dewey, National Library of Medicine, and local call number; type of record; form of reproduction; imprint date; language of the text; and country of publication. Kim reviewed the use of OCLC archival tapes at the University of Lowell Libraries to report the number of new titles added by subject, type of material, intellectual level, and language of the text.[2]

METHODS

To simplify the programming and to store more records on a single tape, the longer, variable-length OCLC records were transformed into shorter, fixed-length master records. Only certain fields were transferred from the OCLC records. The OCLC fields are listed in figure 1.

OCLC Fields Used in Master Record

OCLC Number
Date Cataloged (originally from 001, later from 005 field)
Type of Material
Call Number
Location (from library symbol in 049 field)
Holdings
Title
Author
Edition
Publisher
Series
Physical Description for AV and Sound Recordings
Volumes and Dates for Serials
Subject Headings (limit of two)
007 Field
008 Field

Fig. 1. OCLC Fields.

The information selected from the 007 and 008 fields depended upon the type of material the OCLC record represented. Many fields were truncated. The author field was limited to twenty characters and the title field to seventy-five characters.

New fields were created using information found on the OCLC records. A number of these fields contained flags to identify LC call numbers, records produced during retrospective conversion, monographs over 31 centimeters in height, and women's studies materials. Also included on the master record was a three-digit number representing the university department that ordered the material.

Besides the program to create master records, programs were written to eliminate duplicate and cancelled records, to calculate the collection growth, and to print reports. The programs were written in COBOL to run in a batch mode on an IBM 4341. The program that read OCLC archival tapes and created master records was written by the systems librarian. The remaining programs were written by four

UW–Whitewater undergraduate students enrolled in a two-semester Systems Analysis and Design class. The systems librarian worked closely with the students in designing and writing the programs.

REPORTS

Three reports were produced to help evaluate collection growth by type of material, location, and subject, based upon the LC call number. Growth is represented by the number of titles and the percentages they represent for three time periods: (1) the three months on the OCLC tape, (2) the fiscal year which included the three months, and (3) totals to date.

The type-of-material report is shown in table 1.

The location used in the location report (see table 2) is determined by the fourth character of UW–Whitewater's holding symbol (GZT) in the 049 field.

The LC classification system was subdivided into 423 categories for the subject analysis. An example of this report is shown in table 3.

Subject analysis was also reported by departments. Table 4 represents a subject ordering profile for the Marketing Department. In addition to stating the subject areas in which the department orders materials, the report also gives the percentage of titles the department has ordered, which is compared to all the titles ordered by the library and by other departments in a subject area. For example, the Marketing Department ordered 12.5 percent of all titles acquired in the *HA* LC call-number range.

APPLICATIONS INVOLVING STATISTICAL PACKAGES

The OCLC tape extraction programs can also be used with statistical languages such as SAS, SPSS-X, or Minitab to generate collection management data on request. Such use allows greatly increased flexibility in the production of reports and faster responses to specific questions. Compared with programs written in traditional languages such as COBOL, the statistical packages save considerable amounts of programmer time, require minimal amounts

Table 1. Type of Material Report
Collection Growth Statistics, All Materials, February 7, 1985

Transaction Tape: 04/01/84–06/30/84
Fiscal Year: 07/01/83–06/30/84
To Date: 1976 To Present

	Transaction Tape	Percentage	Fiscal Year	Percentage	To Date	Percentage
Books	11,901	98.4	32,236	97.4	177,743	96.0
Charts	0	0.0	0	0.0	0	0.0
Filmstrips	34	0.3	91	0.3	950	0.5
Games	0	0.0	0	0.0	4	0.0
Instructional Material	2	0.0	20	0.1	198	0.1
Kits	0	0.0	12	0.0	131	0.1
Microforms	0	0.0	1	0.0	8	0.0
Models	0	0.0	0	0.0	0	0.0
Motion Pictures	10	0.1	41	0.1	468	0.3
Musical Scores	60	0.5	237	0.7	1,781	1.0
Others (AV)	0	0.0	3	0.0	18	0.0
Realia	0	0.0	0	0.0	0	0.0
Serials	44	0.4	278	0.8	1,905	1.0
Slides	3	0.0	10	0.0	176	0.1
SD Recordings	17	0.1	79	0.2	1,071	0.6
SD Rec., Nonmusical	17	0.1	53	0.2	647	0.3
Transparencies	0	0.0	3	0.0	33	0.0
Videorecordings	2	0.0	17	0.1	87	0.0
Totals	12,090	99.9	33,081	99.9	185,220	100.0

Table 2. Location Report
Collection Growth Statistics, All Materials, February 7, 1985

Transaction Tape: 04/01/84–06/30/84
Fiscal Year: 07/01/83–06/30/84
To Date: 1976 To Present

	Transaction Tape	Percentage	Fiscal Year	Percentage	To Date	Percentage
Music Center (GZTB)	13	0.1	60	0.2	565	0.3
LMC: Books (GZTC)	33	0.3	256	0.8	3,834	2.1
Documents & Res (GZTD)	91	0.8	877	2.7	8,557	4.6
LMC: AV (GZTF)	69	0.6	256	0.8	2,645	1.4
LMC: Books (GZTL)	2,579	21.3	6,039	18.3	9,772	5.2
Main (GZTM)	9,264	76.6	25,299	76.5	154,360	83.3
Periodicals (GZTP)	0	0.0	0	0.0	1	0.0
Reference (GZTR)	32	0.3	217	0.7	4,331	2.3
Special Cols (GZTS)	3	0.0	31	0.1	777	0.4
LMC: Textbooks (GZTT)	6	0.0	46	0.1	378	0.2
Totals	12,090	100.0	33,081	100.2	185,220	99.9

Table 3. Subject Analysis Report

OCLC Tape for 04/01/84–06/30/84
No. of Titles on Tape: 12,090
Titles for Fiscal Year 07/01/83–06/30/84: 33,081
Titles from 1976 to Date: 185,220

LC Call Number	Subject	3-Mo. Tape Titles	%	Fiscal Year Titles	%	To Date Titles	%
HJ9701	Public Accounting	1	0.0	1	0.0	10	0.0
HM	Sociology	3	0.0	50	0.2	392	0.2
HN	Social Problems & Social Reform	4	0.0	18	0.1	266	0.1
HQ0001	Marriage and Family	23	0.2	98	0.3	615	0.3
HQ1060	Aged & Gerontology	0	0.0	12	0.0	89	0.0
HQ1101	Woman & Feminism	41	0.3	132	0.4	566	0.3
HS	Societies: Secret, Benevolent, Etc.	0	0.0	0	0.0	7	0.0
HT0001	Communities	2	0.0	12	0.0	148	0.1
HT0601	Classes	0	0.0	3	0.0	33	0.0
HT1501	Races	1	0.0	1	0.0	8	0.0
HV0001	Social and Public Welfare	16	0.1	87	0.3	691	0.4
HV6001	Criminology	7	0.1	36	0.1	245	0.1
HV7231	Penology	6	0.0	38	0.1	277	0.1
HX	Socialism	4	0.0	11	0.0	132	0.1
J	Political Science, Official Documents	0	0.0	0	0.0	4	0.0
JA	Political Science, General Works	1	0.0	1	0.0	47	0.0
JC	Political Science, Theory	2	0.0	4	0.0	104	0.1
JF	Political Science, Constitutional History	1	0.0	8	0.0	48	0.0
JK	Political Science—U.S.	3	0.0	27	0.1	195	0.1
JL	Political Science—Latin America, Brit. America, Etc.	0	0.0	1	0.0	18	0.0
JN	Political Science—Europe	4	0.0	12	0.0	103	0.1
JQ	Political Science—Asia, Africa, Australia, Etc.	0	0.0	3	0.0	28	0.0
JS	Political Science, Local Government	0	0.0	3	0.0	40	0.0
JV0001	Colonies and Colonization	0	0.0	0	0.0	4	0.0
JV6000	Emigration and Immigration	1	0.0	1	0.0	12	0.0
JX	Law, International	0	0.0	9	0.0	121	0.1

Table 4. Subject Ordering Profile
Collection Growth Statistics by Department, February 7, 1985—Breakdown by Call Numers

OCLC Tape for 04/01/84–06/30/84
No. of Titles on Tape: 12,090
Titles for Fiscal Year 07/01/83–06/30/84: 33,081
Titles from 1976 to Date: 185,220

460	*Marketing*	*44*	*0.4*
	Fiscal Year Totals:	*95*	*0.3*
	1976 to Date Totals:	*616*	*0.3*

LC Call Number	Subject	3-Mo. Tape Titles	3-Mo. Tape %	Fiscal Year Titles	Fiscal Year %	To Date Titles	To Date %	Subject %
BF001	Psychology	0	0.0	0	0.0	2	0.3	0.2
E0075	Indians of North America	0	0.0	0	0.0	1	0.2	0.5
E0150	History—U.S., General	0	0.0	0	0.0	1	0.2	0.3
F0576	History—Wisconsin	0	0.0	0	0.0	1	0.2	1.6
G0001	Geography	1	2.3	1	1.1	1	0.2	1.4
H	Social Sciences, General Works	0	0.0	0	0.0	7	1.1	5.3
HA	Statistics	0	0.0	0	0.0	8	1.3	12.5
HB	Economic Theory	0	0.0	1	1.1	4	0.6	1.4
HC	Economic Production	5	11.4	10	10.5	71	11.5	5.7
HD0101	Land	0	0.0	0	0.0	1	0.2	0.8
HD2321	Industry	0	0.0	0	0.0	22	3.6	13.2
HD4801	Labor	0	0.0	0	0.0	3	0.5	0.6
HD9000	Special Industries and Trades	1	2.3	5	5.3	26	4.2	9.2
HE	Transportation & Communication	0	0.0	1	1.1	5	0.8	2.2
HF0351	History of Commerce	0	0.0	1	1.1	1	0.2	20.0
HF1001	Commerce, General	1	2.3	3	3.2	19	3.1	67.9
HF1401	Commercial and Tariff Policy	2	4.5	3	3.2	17	2.8	19.3

of code, and possess the flexibility needed to run slightly varying statistical reports without major program rewriting.

Writing in the Winter 1981 *Drexel Library Quarterly*, Charles T. Townley described the use of SAS to analyze a random sample of book collection data,[3] and Jack Slater compared SAS with BASIC in an online collection statistics program.[4] Both authors found that statistical packages could provide faster programming than conventional languages, and Townley provided numerous examples of how a statistical package can manipulate data to provide the desired statistics.

At UW–Whitewater a statistical package has been used in conjunction with the extracted OCLC tapes to provide statistical reports. The fixed-length, relatively short records provided by the extraction programs can be used for data sets for the statistical programs, whereas the bulky, variable-length records on the OCLC tapes cannot. As a result, the enormous statistical capability of programs such as SAS can be used for library analysis of the information contained on the OCLC tapes.

A program run at UW–Whitewater in January 1984 gives an example of the type of information that the statistical packages can provide. The library staff wanted to find out how the age of material affected circulation in a relatively young collection where about 85 percent of the monographic material was less than twenty-five years old. In addition, they were interested in developing a historical profile showing which subject areas had been heavily collected at any given period of time. To obtain this information, they decided to analyze the OCLC tapes from 1976 to June 1982. All retrospective conversion done during this period involved circulating material and could hence be used to approximate circulation statistics.

The figures that the library retrieved were the number of items in the collection published in a given year or range of years, the number of items in each one- or two-letter LC class, and the number and percentage of titles in each call-number class.

The program was written and run in

SAS[5] on the IBM 4341 at UW–Whitewater. The total time between the initial contact of the systems librarian and the university computer center analyst[6] and the finished output was less than one workday, with the actual programming and implementing time running about two to three hours. The final computer run, which plotted LC class letters against the publication dates, took approximately two minutes of CPU time for over one hundred thousand records.

To produce the data, the extracted OCLC tapes from 1976 to June 1982 were mounted and read to produce a data subset containing a one-character type-of-material code, the first two characters of the LC call number, and the date of publication (the first date from the fixed field date in the OCLC fixed fields). The program shown in figure 2 was used to accomplish this.

Next the frequencies of material codes, of call-number letter codes, and of publication dates were calculated using the frequency procedure. An example of the type of output generated for publication dates is shown in figure 3.

Since the printout was capable of displaying twelve date/call-number cells, plus a label and a totals column in each row, twenty-four date ranges were chosen, as follows: before 1900, 1900–09, 1910–19, 1920–29, 1930–39, 1940–49, 1950–59, 1960–66, and single years from 1967 on. This allowed the output to be printed in two sections, one covering the years through 1970 and the other the period from 1971 on. If the second character of the LC call number was numeric, only the first character was included. If the material-type code indicated that the item was anything but a monograph or a serial, the item was not included. The program then performed a frequency procedure that calculated the number of items in each category, the cumulative number of items, the percentage of items in each category, and the cumulative percentage. These statistics were compiled for material type, LC call-number class, publication date range, and for the items in a call-number class published during a certain period (see figure 4).

The results indicated that, although re-

```
FILE: LIBMAST2 SAS     A          VM/SP CONVERSATIONAL MONITOR SYSTEM

DATA LIB.MASTER;
CMS FILEDEF LIBMAST TAP1 (RECFM FB LRECL 360 BLOCK 8260;
INFILE LIBMAST;
INPUT X1 $14. LTYPE $1. LCCALL1 $2. X2 $63. X3 $80. X4 $80. X5 $65.
   DATE1 $4. X6 $51.;
   DROP X1 X2 X3 X4 X5 X6;
OUTPUT LIB.MASTER;
```

Fig. 2. SAS Program Code.

SAS

DATE1	FREQUENCY	CUM FREQ	PERCENT	CUM PERCENT
1932	154	3694	0.140	3.362
1933	145	3839	0.132	3.494
1934	162	4001	0.147	3.641
1935	189	4190	0.172	3.813
1936	210	4400	0.191	4.005
1937	189	4589	0.172	4.177
1938	202	4791	0.184	4.360
1939	202	4993	0.184	4.544
1940	232	5225	0.211	4.755
1941	202	5427	0.184	4.939
1942	241	5668	0.219	5.159
1943	206	5874	0.187	5.346
1944	160	6034	0.146	5.492
1945	201	6235	0.183	5.675

Fig. 3. Publication Date Output.

```
FILE: LIBMAST3 SAS      A                    VM/SP CONVERSATIONAL MONITOR SYSTEM

DATA LIB2;
SET LIB.MASTER;
  X = INDEXC (LCCALL1,'1234567890¢.,!;:','IF'X);
  IF X ¬= 0 THEN SUBSTR(LCCAL 1,X,1) = ' ';
  X = INDEXC (LCCALL1,'1234567890¢.,!;:','IF'X);
  IF X ¬= 0 THEN SUBSTR(LCCALL1,X,1) = ' ';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  IF DATE1 < '1900' THEN DATE2 = ' B4';
  IF DATE1 > '1899' AND DATE1 < '1910' THEN DATE2 = ' 09';
  IF DATE1 > '1909' AND DATE1 < '1920' THEN DATE2 = ' 19';
  IF DATE1 > '1919' AND DATE1 < '1930' THEN DATE2 = ' 29';
  IF DATE1 > '1929' AND DATE1 < '1940' THEN DATE2 = ' 39';
  IF DATE1 > '1939' AND DATE1 < '1950' THEN DATE2 = ' 49';
  IF DATE1 > '1949' AND DATE1 < '1960' THEN DATE2 = ' 59';
  IF DATE1 > '1959' AND DATE1 < '1967' THEN DATE2 = ' 66';
  IF DATE1 > '1982' THEN DATE2 = ' 09';
  IF DATE1 > '1966' AND DATE1 < '1983' THEN DATE2 = SUBSTR(DATE1,3,2);
  IF LTYPE = 'A' OR LTYPE = 'S';
  DROP DATE1;
 PROC FREQ;
  TABLES LTYPE LCCALL1 DATE2 LCCALL1*DATE2;
```

Fig. 4. SAS Program Code.

cent materials circulated more often than older items, even the oldest items in the collection also circulated. Similar brief programs could also be used for other collection management purposes. For instance, the number of majors in a given area or the number of students enrolled in that area could be compared with the number of books purchased in that area. The percentage of the library's total purchases in a given area, either in a specific year or over a period of time, could be readily determined. If a department determined which call number ranges contained material relevant to its program, SAS could then be used to determine the library's strength in supporting the program.

The figures could also be used to suggest areas where deeper analysis might prove fruitful. For example, at UW–Whitewater, more students are enrolled in math courses than in any other department. The number of students in the Management Computer Systems major has grown from 89 in 1978 to 790 in 1983, making it one of the three most popular majors on campus. The number of titles in *QA*, which contains mathematics and computer science, has not kept pace with the rapid growth in computer science and does not reflect either the math enrollment or the increasing importance of computers on the campus at large (see figure 5).

As a result of this preliminary analysis, the Computer Center and the Management Computer Studies (MCS) program were given an additional allocation in 1983–84 to enable them to purchase more titles.[7] Before the 1984–85 allocations became final, however, the extracted tapes were further analyzed. Using the tapes through March 1984, four call number areas were examined: *HF5584* for data processing; *QA76* for computer science; *T58* for management information systems, and *TK5885–5995* for computer engineering. Each area was then broken down by publication date and ordering department fields. Of the twenty nonlibrary departments that bought materials in these four computer science call-number areas, four had purchased over 30 titles. These were Management (214 titles), MCS program (187 titles), Mathematics (126 titles), and Computer Center (79 titles). Of these, Management and Math supply faculty for the MCS program, teach computer-related courses, and are supposed to use some of their library funding to support the computer areas. The MCS program and the Computer Center received relatively small allocations to support their programs, with the understanding that Math and Management would also order materials (see figure 6).

The statistical analysis made it clear that the total number of titles ordered each year remained fairly constant between 1977 and 1984[8], rather than growing with the development of the computer program (see figure 7).

The major reason for the lack of growth was the drop in Management Department purchases, which fell rapidly after 1979. The overall departmental totals reflect this decline, as do the listings for each call number area (see figures 8 and 9).

For the 1984–85 fiscal year, increased allocations for the computer science area will continue, and the library will attempt to determine how this area can best be funded.

The statistical analysis also indicated that the library should carefully study its weeding policies in the computer science areas. The table of publication dates indicated that over half of the titles had been published in or before 1975, and the call number analyses showed a range from 67 percent in *HF5548* to 38 percent in *QA76*. While some of the earlier titles are classics, many are outmoded and present an inaccurate picture to their users.

ENHANCEMENTS TO THE EXTRACTION PROGRAMS

Several enhancements to the existing programs have been considered. One possibility is to include the purchase price in the master record. Using the purchase price, the collection growth reports would state not only the number and percentage of titles but also the amount of collection development funds spent for these titles. Before this can be accomplished, procedures have to be devised for handling serial or multivolume materials and for dealing with materials where the price is out of harmony with either the publication or cataloging date.

Another projected enhancement is to print the material type and location reports by departments. An additional report needs to be developed that would list a subject along with the departments that ordered material in it. By referring to this report a librarian could identify departments that were heavy buyers in a certain subject area. These enhancements could also be

SAS 11:03 FRIDAY, JANUARY 27, 1984

TABLE OF LCCALL1 BY DATE2

LCCALL1 / FREQUENCY / PERCENT / ROW PCT / COL PCT	DATE2 71	72	73	74	75	76	77	78	79	80	81	82	TOTAL
Q	15 0.01 4.75 0.30	14 0.01 4.43 0.24	7 0.01 2.22 0.14	20 0.02 6.33 0.45	18 0.02 5.70 0.37	8 0.01 2.53 0.12	17 0.02 5.38 0.26	25 0.02 7.91 0.34	30 0.03 9.49 0.46	15 0.01 4.75 0.29	4 0.00 1.27 0.14	2 0.00 0.63 0.48	316 0.31
QA	134 0.13 4.62 2.71	144 0.14 4.96 2.46	170 0.16 5.86 3.41	260 0.25 8.96 5.83	301 0.29 10.37 6.22	199 0.19 6.85 3.09	226 0.22 7.79 3.48	208 0.20 7.17 2.82	182 0.18 6.27 2.80	147 0.14 5.06 2.80	82 0.08 2.82 2.86	5 0.00 0.17 1.20	2903 2.81
QB	15 0.01 4.05 0.30	21 0.02 5.68 0.36	26 0.03 7.03 0.52	19 0.02 5.14 0.43	25 0.02 6.76 0.52	20 0.02 5.41 0.31	7 0.01 1.89 0.11	10 0.01 2.70 0.14	8 0.01 2.16 0.12	9 0.01 2.43 0.17	4 0.00 1.08 0.14	2 0.00 0.54 0.48	370 0.36
QC	26 0.03 4.41 0.53	35 0.03 5.93 0.60	15 0.01 2.54 0.30	36 0.03 6.10 0.81	54 0.05 9.15 1.12	38 0.04 6.44 0.59	21 0.02 3.56 0.32	21 0.02 3.56 0.28	32 0.03 5.42 0.49	11 0.01 1.86 0.21	10 0.01 1.69 0.35	3 0.00 0.51 0.72	590 0.57
QD	26 0.03 5.35 0.53	28 0.03 5.76 0.48	34 0.03 7.00 0.68	30 0.03 6.17 0.67	33 0.03 6.79 0.68	34 0.03 7.00 0.53	26 0.03 5.35 0.40	37 0.04 7.61 0.50	24 0.02 4.94 0.37	12 0.01 2.47 0.23	5 0.00 1.03 0.17	0 0.00 0.00 0.00	486 0.47
QE	9 0.01 2.86 0.18	15 0.01 4.76 0.26	8 0.01 2.54 0.16	18 0.02 5.71 0.40	22 0.02 6.98 0.45	32 0.03 10.16 0.50	26 0.03 8.25 0.40	23 0.02 7.30 0.31	25 0.02 7.94 0.38	14 0.01 4.44 0.27	2 0.00 0.63 0.07	1 0.00 0.32 0.24	315 0.30
QH	34 0.03 4.47 0.69	63 0.06 8.29 1.07	47 0.05 6.18 0.94	50 0.05 6.58 1.12	53 0.05 6.97 1.09	65 0.06 8.55 1.01	61 0.06 8.03 0.94	42 0.04 5.53 0.57	24 0.02 3.16 0.37	38 0.04 5.00 0.72	10 0.01 1.32 0.35	4 0.00 0.53 0.96	760 0.73
QK	10 0.01 2.69 0.20	21 0.02 5.65 0.36	12 0.01 3.23 0.24	18 0.02 4.84 0.40	19 0.02 5.11 0.39	17 0.02 4.57 0.26	15 0.01 4.03 0.23	20 0.02 5.38 0.27	21 0.02 5.65 0.32	22 0.02 5.91 0.42	4 0.00 1.08 0.14	3 0.00 0.81 0.72	372 0.36
QL	38 0.04 4.14 0.77	43 0.04 4.68 0.73	38 0.04 4.14 0.76	48 0.05 5.23 1.08	51 0.05 5.56 1.05	50 0.05 5.45 0.78	41 0.04 4.47 0.63	31 0.03 3.38 0.42	45 0.04 4.90 0.69	48 0.05 5.23 0.91	18 0.02 1.96 0.63	3 0.00 0.33 0.72	918 0.89
TOTAL	4942 4.78	5864 5.67	4985 4.82	4463 4.31	4841 4.68	6436 6.22	6496 6.28	7377 7.13	6501 6.29	5254 5.08	2865 2.77	418 0.40	103436 100.00

(CONTINUED)

Fig. 5. *Sample LC Call Number/Date Output—Q–QL, 1971–1982.*

```
FILE: LIBMAST7 SAS        A1                 VM/ P CONVERSATIONAL MONITOR SYSTEM


DATA LIB2;
SET LIB.MASTER LIB2.MASTER;
  X = INDEXC (LCCALL1,'¢,!;:','1F'X);
  IF X ¬= 0 THEN SUBSTR(LCCALL1,X,1) = ' ';
  X = INDEXC (LCCALL1,'¢,!;:','1F'X);
  IF X ¬= 0 THEN SUBSTR(LCCALL1,X,1) = ' ';
  LCCALL1 = COMPRESS (LCCALL1);
  IF SUBSTR (LCCALL1,1,4) = 'QA76' THEN LCCALL1 = SUBSTR (LCCALL1,1,4);
  IF SUBSTR (LCCALL1,1,3) = 'T58' THEN LCCALL1 = SUBSTR (LCCALL1,1,3);
  IF LCCALL1 = 'QA76'
   OR LCCALL1 = 'HF5548'
   OR (LCCALL1 >= 'TK7885' AND LCCALL1 <= 'TK7895')
   OR LCCALL1 = 'T58';
  IF LTYPE = 'A' OR LTYPE = 'S';
  IF LCCALL1 >= 'TK7885' AND LCCALL1 <= 'TK7895' THEN LCCALL1 = 'TK78--';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  X = INDEXC (DATE1,'U');
  IF X ¬= 0 THEN SUBSTR(DATE1,X,1) = '0';
  LENGTH DATE2 $ 6;
  IF DATE1 > '0000' AND DATE1 < '1975' THEN DATE2 = '190074';
  IF DATE1 > '1984' THEN DATE2 = '190074';
  IF DATE1 > '1974' AND DATE1 < '1985' THEN
        DATE2 = SUBSTR(DATE1,1,4) || SUBSTR(DATE1,3,2);
  DROP DATE1;
 PROC FREQ;
  TABLES LTYPE DEPT LCCALL1 DATE2 LCCALL1*DATE2 DEPT*DATE2
      LCCALL1*DEPT*DATE2;
```

Fig. 6. Listing of the Program for the Statistical Analysis.

SAS

DATE2	FREQUENCY	CUM FREQ	PERCENT	CUM PERCENT
	4	.	.	.
190074	880	880	45.082	45.082
197575	139	1019	7.121	52.203
197676	125	1144	6.404	58.607
197777	132	1276	6.762	65.369
197878	139	1415	7.121	72.490
197979	115	1530	5.891	78.381
198080	113	1643	5.789	84.170
198181	139	1782	7.121	91.291
198282	94	1876	4.816	96.107
198383	72	1948	3.689	99.795
1984 84	4	1952	0.205	100.000

Fig. 7. Total Number of Titles Ordered in Computer Science, by Year of Publication.

provided through the use of SAS or some other statistical package, as described above.

In addition to producing the collection growth reports, the master records were also designed to produce bibliographies and statistics on demand. Using the fields on the master record that are not available or easily accessed in the card catalog, bibliographies with brief, truncated information could be printed. Bibliographies could be limited by type of material, publication date, subject, and ordering department as well as by other information stored on the master record. For instance, bibliographies covering audiovisual materials concerning the Equal Rights Amendment or listing recently acquired reference materials could be generated. It would also be possible to print the titles of all materials written or performed in Spanish or some other language. By the appropriate choice of parameters, a wide range of bibliography types could be produced.

In the future, the library is planning to expand its use of statistical packages to analyze the OCLC tapes. When new courses are proposed, SAS or one of its cousins will enable the library to look at specific call-number groupings to determine how much collection support exists for the course, how current the materials are, and which departments are ordering the materials. The selection of audiovisual materials can be examined more closely than has previously been possible. More detailed breakdowns of departmental spending can be made to ensure that a well-rounded collection is developed.

The potential of SAS for library applications, using OCLC extracted tapes for input, goes far beyond the frequency tabulations described here. SAS is capable of performing highly complex, multivariate procedures as well as simple descriptive statistics. By selecting the appropriate fields for an OCLC extraction tape, librarians could perform a wide variety of statistical analyses. SAS programs using extracted tapes could be used to supplement and enhance the statistics available from online library systems and to provide individually tailored statistical reports. Through SAS/Graph,[9] output could be produced in a color graphics format suitable for visual presentations.

CONCLUSION

A program for extracting OCLC tapes enables a library to select and abbreviate fields from its OCLC tapes and create a machine-readable record that can be used for many purposes. The programs can provide printouts of information that will need to be produced regularly, such as the collection growth reports generated at UW–

SAS 13:35 MONDAY, OCTOBER 8, 1984

TABLE OF DEPT BY DATE2

DEPT / FREQUENCY / PERCENT / ROW PCT / COL PCT	DATE2	190074	197575	197676	197777	197878	197979	198080	198181	198282	198383	198484	TOTAL
450	0	12	3	25	59	53	10	4	20	11	14	0	211
	.	0.95	0.24	1.99	4.69	4.22	0.80	0.32	1.59	0.88	1.11	0.00	16.79
	.	5.69	1.42	11.8	27.96	25.12	4.74	1.90	9.48	5.21	6.64	0.00	
	.	2.68	13.04	62.50	72.84	38.97	8.70	3.64	14.81	11.70	19.72	0.00	

Fig. 8. Management Department Purchases in Computer Science, by Year of Publication.

SAS 13:35 MONDAY, OCTOBER 8, 1984

TABLE OF DEPT BY DATE2
CONTROLLING FOR LCCALL1=QA76

DEPT / FREQUENCY / PERCENT / ROW PCT / COL PCT	DATE2	190074	197575	197676	197777	197878	197979	198080	198181	198282	198383	198484	TOTAL
450	0	7	1	18	42	34	6	1	14	3	2	0	128
	.	0.92	0.13	2.36	5.50	4.45	0.79	0.13	1.83	0.39	0.26	0.00	16.75
	.	5.47	0.78	14.06	32.81	26.56	4.69	0.78	10.94	2.34	1.56	0.00	
	.	3.76	7.69	64.29	77.78	32.38	6.59	1.30	13.73	4.76	4.76	0.00	

Fig. 9. Management Department Purchases—QA76 Only, by Year of Publication.

Whitewater. For those libraries with access to a statistical package, SAS or one of its cousins—used in conjunction with an extracted tapes program—provides a wide range of possibilities for analyzing library statistics, especially in the area of collection management. By selecting the appropriate data sets, procedures, and tables, a library can obtain flexibility far beyond that provided by programs in a standard language such as COBOL. Use of both the regular reports generated by the extraction programs and the special purpose reports generated by the statistical program provides librarians with a range of options for generating statistics. These statistical reports can either be used by themselves for collection management or serve as a supplement to the reports generated by an online system.

REFERENCES AND NOTES

1. Glyn T. Evans, Roger Gifford, and Donald R. Franz, *Collection Development Analysis Using OCLC Archival Tapes; Final Report* (U.S. Department of Health, Education, and Welfare, Office of Education, 1977).
2. David U. Kim, "OCLC-MARC Tapes and Collection Management," *Information Technology and Libraries* 1:22–27 (Mar. 1982).
3. Charles T. Townley, "Using SAS to Analyze Book Collection Data," *Drexel Library Quarterly* 17:61–74 (Winter 1981).
4. Jack Slater, "Online Collection Statistics: a Comparison of BASIC and SAS," *Drexel Library Quarterly* 17:87–119 (Winter 1981).
5. *SAS Users Guide: Basics, 1982 Edition* (Cary, N.C.: SAS Institute, 1982).
6. Atlee Svanoe of the UW–Whitewater Computer Center was the system analyst responsible for the SAS code in these programs.
7. The number of computer science periodicals purchased has also increased significantly over the past two years. Increasing the book funds has been only a part of the library's efforts to strengthen the collection in this area.
8. The seeming drop in titles from 1982 on is a result of the time lag between when a title is published and when it is ordered and cataloged.
9. For information about SAS/Graph, see the *SAS/Graph User's Guide, 1981 Edition* (Cary, N.C.: SAS Institute, 1981) and *SAS Views: SAS Color Graphics, 1983 Edition* (Cary, N.C.: SAS Institute, 1983). ■■

Communications

Analysis of Survey Results Using a Microcomputer

Elena Romaniuk

INTRODUCTION

In response to the Decentralized Plan for Canadian Newspaper Preservation and Access announced by the National Library of Canada in 1983,[1] the Newspapers Committee of the British Columbia Library Association (BCLA) prepared a document that is to serve as a plan for the British Columbia Newspaper Project. The definition of the project states that it is to be "a province-wide plan for cooperative bibliographic and holdings control and preservation of and access to British Columbia newspapers."[2]

In order to get an overview of the general disposition of newspapers within the province and to compile a list of known repositories, the BCLA Newspapers Committee conducted a preliminary survey of newspaper collections held by British Columbia (B.C.) libraries, museums, and archives. A short questionnaire was sent out to approximately three hundred institutions throughout the province.

This communication describes how the results of the survey were analyzed using a microcomputer and commercially available software. The analysis is applicable to many library surveys.

EQUIPMENT

The hardware consisted of an Apple IIe computer with 128K bytes of memory and an 80-column card, two Apple disk drives, a Comrex amber monitor, and a Gemini-10X printer.

PFS: file and *PFS:report* were the two commercially available software packages used to create and manipulate the database. The information was stored on single-sided, double-density floppy diskettes.

This combination of hardware and software was used for several reasons. The computer equipment was available in this particular configuration, and it was anticipated that a reasonably small database of approximately 300 records would be created. It did not seem likely that the size of the database would require the use of a hard disk drive, which would provide more storage and facilitate faster access to records in a large database, but which would not make a significant difference in this case. PFS software was chosen because it is compatible with the hardware configuration used, it is relatively inexpensive and widely available commercially, and it is very easy to use. The information stored in the file can be retrieved in a variety of ways. It can be updated, printed, or manipulated using other PFS software. The user requires no programming skills to perform any of these functions, as is often the case with other file management software.

METHODOLOGY

The analysis of survey results was achieved in two parts. First, a database was created using *PFS:file*. This was then used to generate reports of the available data with *PFS:report*.

The questionnaire prepared for the survey was short, consisting of the following four questions:

1. How many B.C. newspaper titles are you currently receiving on subscription or as a gift?
2. How many backfiles of B.C. newspaper titles in original letterpress form do you have in your collection?
3. How many newspaper titles in microform do you have in your collection?

Elena Romaniuk is serials librarian, McPherson Library, University of Victoria, Victoria, B.C., Canada.

4. Do you have any special individual issues (anniversary, etc.) of B.C. newspapers in your collection?

The first three questions required a numerical answer, the last either yes or no. In addition, the person answering the questionnaire was to include his or her name and job title, the name of the institution, phone number, and the date. This information, together with the mailing lists of institutions to which the questionnaires were sent, was used to create the database.

Creating the Database

PFS:file can be used without any previous exposure to the Apple IIe. It is menu driven, consisting of screens containing numbered choices for each option, from which the user selects the appropriate number depending on the function the computer is to perform. Each menu is clearly presented and the choices given are unambiguous. Included also are warning screens that give the user the option to abandon the function in progress without change or loss of data. *PFS:file* is a program designed to organize and store information in records that are created by the user for a specific purpose. The record consists of a number of fields, the organization and order of which are determined by the user. The number of spaces allowed for each field is not fixed, so character strings of variable length can be input. The software is designed to compress the records where the fields are not completely filled with relevant data. Blank spaces are excluded from the record while it is stored on the diskette, so only the minimum amount of storage space is used up.

There are a number of ways to retrieve information from the file. The fastest way to search the file is to look for information in the first field. The design of the software allows very fast access to the first field by going directly to the relevant record, without needing to search through the whole file; however, one can search for any field in each record, as well as a combination of fields. Searching for a combination of fields within a record is equivalent to Boolean *and* searching, but the equivalent of *or* is not available. As an example, one can retrieve all institutions that answered all three questions with nine or more titles, but one cannot determine, in one step, which institutions hold nine or more titles in the first, second, or third category. To get the answer, one must search the file three times, once for each of the three questions. When searching for information not contained in the first field, the program systematically searches the whole file by looking at each record.

Various options are available for searching for an item within a field. One way to search is by complete item match. This means that only those records that exactly match the items being searched in given fields will be retrieved. It is also possible to search for a partial item match by telling the program to ignore characters before or after a specific character string. This is equivalent to searching for a key word within a field. There are also several ways to search for numbers: one can search for a full item match disregarding the numerical value of the data; for all items greater than, less than, or equal to a certain number; or for numeric values within a range. Finally, it is also possible to reverse the intent of any search using the *not* match. In addition, the program does not distinguish between upper- and lowercase letters. It treats multiple spaces between characters as one space and it ignores spaces before the first character and after the last. These features make file searching even more flexible.

In designing the record for this file, fourteen fields were used. Those relating to the institution were labeled as: name; address; city; pc (postal code); contact; phone; geo (geographic code); and type of library. Also included was the date the answered questionnaire was received. The fields for the answers to the questions were labeled as: current (question 1); orig bfile (question 2); micr bfile (question 3); and special (question 4). Finally, a field labeled "comments" was included to allow the addition of miscellaneous information. To facilitate fast access by name of the institution, it was positioned as the first field in the record.

Initially the database was created using the *add* function, by input of the names and addresses of institutions to which the questionnaires were sent. A total of 299 records was filled in. In order to input the information provided by the returned questionnaires, each institution's record had to be retrieved from the file by name and updated. This was done using the *search/update* function. In updating the database, it was found that institutions quite often did not identify themselves with the same name as the one entered from the mailing list. Names were often shortened or changed, so it was not always possible to search the file by full item match in the first field. In these cases partial item searching proved very useful.

While working with the database, a backup of the file was always made. This was done in case the original was either damaged or lost. Copies of the file were made using the *copy* function. This function is also useful for splitting files that have become too large or for merging files from different diskettes.

PFS:file functions discussed so far were ones most frequently used in building the database. However, the program offers two other major functions. These are the *print* and *remove* functions. By selecting the option to print in the main menu, one can get hard copy of the information in the file without using additional PFS software. One can choose to print the whole file, just one record, or a subset of fields in a record. Before printing, one can choose a field that will be used to sort the information in the file. By selecting the *remove* function one can get rid of records no longer wanted in the file.

Generating reports

PFS:report is a program designed to produce reports in the form of tables from existing PFS files. This program is also very easy to use, requiring little experience with a computer. It is menu driven, providing the user with three clear choices in the main menu. These are

1. Print a report.
2. Predefine a report (used to save a report design for repeated use).
3. Set new headings (used to change name headings of fields printed in the report).

As in *PFS:file*, each function is also menu driven. The user is prompted to choose numbers corresponding to the desired action or to fill in information required for searching and output.

Once the function to print a report is chosen, three steps are needed to complete the process. In the first, the user is required to indicate which records are to be retrieved. Retrieval specifications are filled in exactly as in *PFS:file*. The same choices for searching are available: the full item match, the partial item match, numeric matches and the *not* match. Spaces and upper- and lowercase characters are treated as in *PFS:file*. In the second step, the report title is filled in and predefined report specifications can be given. In the third step, the user determines which fields from the records are to appear as columns in the report and in what order. The program determines how many spaces are required for each column and how the report is printed. If the number of characters to be printed exceeds the available width of the page, the program gives the user a choice of abandoning the report or printing it in a truncated form. Before output, the program also sorts the data in the first column into alphabetical or numerical order. If the information in the first column is identical, then the program sorts the second column. A diskette called Sortwork is provided with the software to facilitate the sorting.

A number of different reports were generated from the database[3]. One consisted of an alphabetical list of respondent institutions and their holdings. It was produced in five columns, with the name of the institution in the first and the holdings information in the remaining four columns. Each column was labeled using the appropriate field name from the file, but it is also possible to change field name headings for the purpose of printing a report. The changes will not appear in the file itself, only in the printed report.

The report required only one field to be specified for retrieval. When the database was updated to include information from the responses, the date received was always input. This field was then used in the retrieval specifications menu to search for respondent institutions, since one can search the file for a field in which any characters appear. Similarly, a list of nonrespondent institutions was printed by using the *not* feature with the same retrieval specification. In this case, only those institutions with no data entered in the date-received field were retrieved.

Other reports generated from the database summarized institutions holding backfiles of ten or more current titles, in original and in microform. In each case, the appropriate field was searched by entering >9 in the search specification step. In these reports the first column consisted of the three geographic codes used to indicate the location of each institution. Within each location, the names of the institutions were arranged alphabetically.

A large variety of reports can be generated using *PFS:report*. They can be produced quickly, but have to be printed immediately, since there is no option available to save the report to a disk. Also, if the printing has to be interrupted, usually due to printer malfunction, it cannot be restarted again. To print the same report, one has to start again at the main menu.

CONCLUSION

When the British Columbia Newspaper Project starts the implementation phase, the database created from the survey, together with *PFS:file* and *report* will be used to follow up on recommendations suggested in the plan.[1] Smaller files will be created for the institutions to be contacted again, and together with the PFS word processing program will be used to generate new surveys, form letters, and mailing labels. Information on collection and conservation policies will be gathered, and the physical condition of original newspaper files will be determined. Nonrespondent institutions and those holding special issues will be contacted again. New surveys will be used to gather information on all B.C. newspaper publishers and microfilming agents. The methodology described in this paper will also be used to institute a pilot study on the need to improve access to B.C. newspapers in various regions of the province.

The variety of future applications demonstrates the flexibility of the software and indicates that it could be useful in many library projects.

REFERENCES

1. "National Library Advisory Board Endorses Newspaper Action Plan," *National Library News* 15, no.5-6:5 (May/June 1983).
2. BCLA Newspapers Committee, Plan for the British Columbia Newspaper Project (Victoria, B.C.: British Columbia Library Association, July 1984), p.1.
3. Ibid., A2–A10.
4. Ibid., p.26. ■■

The DELTA Center Concept: A Modest Proposal for the Improvement of Research Library Infrastructure

Norman Howden and Bert R. Boyce

INTRODUCTION

Current success in the development of integrated library computer systems coupled with new economies in communications suggest that a new look at centralization issues in academic libraries may be warranted. Distributed access to library files is made more feasible by automation, and library users are more conditioned to expect remote access to information. A concept for utilizing remote access with a centralized resource is proposed.

The advent of computerization has enabled the librarian to provide a unification of records of the books, periodicals and other material in his library. With computer produced microfiche he can place in each department a set of catalogue entries for the resources available anywhere on the campus and, if he can afford it, a terminal to access the total library database which could in a fully automated system give information not only on books in stock but also on orders, books out on loan, works at the binders etc. When a reader has

Norman Howden is associated with the Center for Wetland Resources, Louisiana State University, and Bert R. Boyce is a member of the faculty of the School of Library and Information Science, Louisiana State University, Baton Rouge.

such facilities, the physical distribution of the library is of less consequence to him.[1]

There is a long-standing debate in academic library circles as to whether it is better to divide library resources into multiple branches, clusters, and professional units or to centralize in a single central facility.[2,3] Debate over branch libraries is further warmed by the sometimes heated discussions on departmental book collections and separately funded special information centers.[4,5] We do not intend to address these problems in the traditional sense of administrative efficiency or user effectiveness, but happily refer the interested reader to those papers cited above. We can see considerable merit on both sides of the debate.

Nonetheless, we are aware of several studies of user behavior that argue strongly for keeping library services as geographically close to users as possible. In 1967, Victor Rosenberg[6] demonstrated that user opinion validated earlier studies of user behavior (e.g., Allen)[7] by showing that preference for an information source was primarily a function of proximity. Also in 1967, Mary Lee Bundy found, in surveying users of a metropolitan public library, that:

39.7 percent traveled less than one mile; 44.4 percent between 1 and 5 miles; 9.1 percent between 10 and 15 miles; 1.8 percent over 15 miles, and 1.5 percent did not respond.[8]

Zweizing and Dervin in 1977, summarizing ten studies measuring users' distance from the public library, found that seven "relate distance negatively to the amount of library use; and three found the relationship nonsignificant."[9] Zweizig's own study suggests that distance may have no relationship to very high or very low use, but a strong relationship to moderate use.[10] Frustrated use, at least in academia, does not seem to translate into no use according to MacVean's 1981 research report, which shows no relationship between library circulation and distance to a faculty member's office.

Interestingly, the modern electronic library with the possibility of remote access to its files provides an alternative to travel. If telecommunications are properly arranged, any authorized patron with a terminal, modem, and telephone can access and search an online library catalog, and perhaps could confirm circulation status or even place materials currently circulated to other users on hold. Such a patron could also search commercial databases, providing he or she is able to afford not only access fees but training and the rather expensive search manuals. This means that much of the work for which the patron might travel to the library can, in fact, be done from the home or office. If arrangements for delivery of required materials could be established, the need for proximity to the library might well be met without confronting the issue of creating branch libraries for patron convenience.

There are at least two problems with an unsupported remote-access concept for users. The first has to do with the cost of commercial databases, the second with the identification of ease of use through proximity alone. Costs for commercial database use are best distributed when documentation is centralized, with a need to buy only a few sets, and when it is necessary to train only a few people to do searching. The best discounts for search time are also obtained for large-volume users, i.e., a centralized account. The second problem is that while individual users may access the library catalog electronically, proximity of access is not the only component of that most important criterion, "ease of use." If the patron finds the methods necessary to use the information system onerous or time-consuming, physical proximity is not enough. Other factors such as terminal availability, terminal compatibility, and user proficiency in an unaided environment become important. If an analogy can be drawn from end users' searching behavior in commercial databases to such patrons' use of library files, we cannot anticipate overwhelming direct use by library patrons using remote terminals.

Walton and Dedert, in reporting the results of training Exxon researchers to conduct their own online literature searches, found that only 20 to 40 percent do any searching after the training sessions despite the fact that these were held for volunteers who expressed a desire to conduct their own searches. Their comments as to why this was so are illuminating:

> Many of our course participants were surprised, indeed discouraged, by the careful thinking and logic that must be applied to do an effectual search. . . . The realization . . . that online searching was a . . . skill requiring time and effort to master was enough to convince them not to search, despite their original intentions.[12]

Haines reports one of six trainees still searching in one group after six months, but seven of fifteen in another group after eleven months.[13] Richardson's figures for continued searching by engineers trained in searching are not more encouraging. Of twenty trained, six remained active searchers after a year.[11]

Thus we believe that while the library can be reasonably accessed electronically from remote locations by the patron, most patrons will prefer the assistance of trained intermediaries (information specialists) who have expert knowledge of the files and their use. Ease of use implies not only physical proximity to information resources but expert assistance and the time to become proficient in the means of access. Ease of use implies not only easy availability of information about the existence and location of desired material but easy availability of the material itself.

If a research library wishes to maintain a centralized collection for the purposes of efficiency, but still desires to build circulation and stimulate use by making its services more easily accessible, that capability certainly exists today. Almost all libraries that have automated their central files can be remotely accessed, and the electronic library that can be remotely accessed provides a measure of distributed service. If assistance to users could be distributed as well, improvements in ease of use without adverse costs could be obtained.

We propose that the Distributed Electronic Library will provide Terminal Assistance. The DELTA Center is conceived as a location manned by a librarian remote from the main library but equipped with the electronic and telecommunications equipment and skills necessary to query the files of the library, commercial databases, and bibliographic utilities and to accomplish interlibrary loan. The DELTA Centers should also be able to request and receive one-day delivery of any materials available locally; these would be picked up by the patron at the center.

The DELTA Center can be physically quite small and probably need only be open during normal working hours. The librarian on duty can provide assistance to users of the local library catalog, perform commercial searches with minimum delay, place books on immediate hold or recall, and even forward reference questions to the central staff.

The number and location of DELTA Centers needed on a campus with a strong research commitment is a matter for further investigation. Ideally a center would be within easy walking distance of every faculty office and residence hall. Perhaps, more practically, they should be located near or in principal research centers or near clusters of research faculty. A wide spectrum of implementation patterns may exist. Present library branches, special information centers, and departmental units may serve as DELTA Centers with the addition of only a terminal, a librarian, or communication facilities. To tie together DELTA Centers, one or more coordinators working from the central library should administer uniform billing, training, and operational procedures. Operational characteristics will vary, of course, as libraries vary in having search services independent of reference activities, search services in library subject units, etc.

An initial implementation of the DELTA Center concept has been proposed for the Louisiana State University at Baton Rouge. If accepted, initial efforts will be devoted to studying the demographic, economic, and organizational factors that enter into decisions for implementing these centers. Results will be reported as rapidly as feasible.

REFERENCES

1. K. W. Humphries, "Centralization and Decentralization in University Libraries," *Australian Academic and Research Libraries* 12:1–6 (Mar. 1981).
2. J. Michael Bruno, "Decentralization in Academic Libraries," *Library Trends* 19:311–17 (Jan. 1971).

3. Ralph E. Russel, "Branch Library Policy Statement," *The Library Scene* 3:28–29 (Sept. 1974).
4. David C. Genaway and Edward B. Stanford, "Quasi-Departmental Libraries," *College & Research Libraries* 3:187–94 (May 1977).
5. Robert P. Haro, "Research Unit Libraries as Special Libraries on the Campus," *Special Libraries* 59:634–37 (Oct. 1968).
6. Victor Rosenberg, "Factors Affecting the Preference of Industrial Personnel for Information Gathering Methods," *Information Storage and Retrieval* 3:119–27 (July 1967).
7. Thomas J. Allen, *The Utilization of Information Sources During R & D Proposal Preparation*, Report no. 97-64 (Cambridge, Mass.: Alfred P. Sloan School of Management, MIT, Oct. 1964).
8. Mary Lee Bundy, "Metropolitan Public Library Use," *Wilson Library Bulletin* 41:950–61 (May 1967).
9. Douglas L. Zweizig and Brenda Dowin, "Public Library Use, Users, Uses: Advances in Knowledge of the Characteristics and Needs of the Adult Clientele," in *Advances in Librarianship* (New York: Academic, 1977), p.231–54.
10. Douglas L. Zweizig, "Predicting Amount of Library Use: an Empirical Study of the Role of the Public Library in the Life of the Adult Public" (unpublished diss., Syracuse University, 1973).
11. Donald S. MacVean, *Faculty Use of a University Circulating Book Collection* (Bethesda, Md.: ERIC Document Reproduction Service, ED 157 561, 1981).
12. Kenneth R. Walton and Patricia Dedert, "Experiences at Exxon in Training End-Users to Search Technical Databases Online," *Online* 7:42–50 (Sept. 1983).
13. Judith Haines, "Experience in Training End-User Searchers," *Online* 6:14–23 (Nov. 1983).
14. Robert J. Richardson, "End User Online Searching in a High Technology Engineering Environment," *Online* 5:44–57 (Oct. 1981).

■■

Reports and Working Papers

MARC Compatibility: A TESLA Survey of Vendors

Dorothy S. McPherson

INTRODUCTION

The results of a 1984 TESLA survey on the MARC compatibility of automated library systems are summarized. Some vendor comments are reported in order to highlight standards issues, and a list of suggested questions, which might be posed to vendors to determine how fully the MARC format has been implemented in their systems, is included.

OVERVIEW

The phrase "MARC compatible" is widely, if imprecisely, used to describe automated systems that meet established national standards for treatment of bibliographic and authority data.[1] The Technical Standards for Library Automation Committee (TESLA) of ALA became concerned several years ago about the extent to which automated library systems were really MARC compatible. With the approval of the LITA board, a questionnaire was sent to 106 systems vendors and networks asking about their compliance with established standards (see appendix A for the text of the questionnaire).

IMPORTANCE OF THE MARC FORMAT

When a library evaluates an automated system, concern about record formats may take a back seat to other criteria such as system features and purchase price. In the long term, however, use of a system that does not meet existing standards may prove extremely costly.

Dorothy S. McPherson is a principal ADP analyst for the Division of Library Automation, University of California, Berkeley.

The current pace of hardware and software development dictates a limited life span for most automated systems. Libraries must anticipate that any system they buy or develop eventually will be replaced. Special translation programs will be required to transfer data from one system to another unless the systems share a common standard data format. In the worst case it may be necessary to manually rekey data that once was present but was not output from the initial system. The ability to move bibliographic data from one system to another is a primary reason for insisting on standardization.

Movement of bibliographic data between institutions is likely to increase in the future as linked system protocols are adopted. Here again, use of the MARC format will be a key requirement.

Standardization of data is also necessary if libraries are to use the "off the shelf" software packages that are increasingly being developed for bibliographic data.

THE SURVEY

The TESLA questionnaire asked whether automated systems were capable of accepting, storing, and outputting MARC format records. Specific questions were asked about the treatment of fixed fields, variable fields, character sets, and extra-length records.

It was gratifying to see a strong commitment to the MARC format on the part of many vendors who responded to the survey. However, there were enough nonstandard practices reported to indicate that MARC compatibility cannot be assumed and that customers should question prospective vendors carefully in a number of areas.

The response rate on the questionnaire was 45 percent—48 returned out of 106 distributed. Although there is no evidence to suggest it, vendors who supported the MARC format may have been more likely to respond and this might make the results unrepresentative. For this reason, and because interpretations of some questions varied, the results of the survey may not be statistically valid. However, the extensive comments of respondents provide valuable insight into the areas that customers should carefully examine.

The following sections outline the responses received and include samples of the comments made by respondents.

USE OF THE MARC FORMATS

Vendors were asked whether their system presently accepted, stored, and output MARC records for books, film, manuscripts, maps, music, and serials. Those who answered "no" for all formats were asked to explain the format used in their system.

Results for each MARC format were tallied separately. Positive responses ranged from a high of 85 percent for books to a low of 65 percent for manuscripts and maps. A "no" response sometimes meant that the system simply did not attempt to support the format in question or that the system supported it but did not use the MARC record structure. Comments from those responding "no" showed that it is common for systems to accept MARC records but to alter record format in their internal processing systems. Sometimes the result is an inability to recapture and output a standard MARC record. The following are typical of comments received from those not supporting the MARC format:

- Our system accepts Marc formats, but does not store and output Marc records in their entirety.
- Output is in [system Y] format.[2]
- Our system will accept (but does not store or output) all MARC record formats. We use OCLC and RLG as our MARC-format record store.

A vendor who basically supported MARC gave a more detailed response:

- In general, the four indicated formats are supported by [vendor X]. However, [vendor X] has not implemented selected fields or subfields which are defined in the USMARC formats but are not of use to [system Y] participants and are not used/distributed by LC, e.g., 017, 023. We add such fields as needed, e.g., when local need arises, or we load records from an external source which uses them.

FIXED FIELDS

Vendors were asked, "Does your system presently accept, store and output all fixed fields from all MARC formats?" Forty-six responses were received to this question, with 72 percent responding "yes" and 28 percent responding "no."

Those who answered "no" were asked to explain any limitations concerning fixed fields in their system. The explanations showed that vendors varied in their support of fixed-field data. For example:

- The system will accept fixed fields identified by the customer as relevant, but does not store or output the associated MARC tags. The data itself is stored and output in a user-defined tag in our system.
- Library choice is to use only 001 & 008.
- Our system maintains only the fixed fields needed for processing catalogs—printed, microfilm and microfiche.
- We could handle all fixed fields. We do only handle [three of them, as they] are the only ones that have been encoded with any reliability over the years (since 1972).

VARIABLE FIELDS

The following question was asked about variable fields: "Does your system presently accept, store, and output all variable field tags from all MARC formats?" Positive responses were higher for variable fields (80 percent) than for fixed fields (72 percent). A "yes" sometimes referred only to those formats supported by the system rather than to all MARC formats.

Positive responses were the same (80 percent) for a second question asking if systems could presently accept, store, and output all variable field indicators. All subfield delimiters for variable fields were also supported by 80 percent of those responding to the question.

Respondents were asked to explain in detail the limitations of their systems if they responded negatively to any of the questions about support for variable fields. Typical comments were

- Our system accepts tags, indicators

and subfield delimiters but does not store or output them. Data in MARC format is read and converted to our system format according to specifications of customer approved conversion programs. Indicators are read and acted upon, that is, non-filing characters are removed according to indicator directions.

• Only a subset of tags are utilized; indicators & subfield delimiters are stripped out.

• System accepts, stores and outputs all tags, indicators and delimiters but only interprets filing indicators and 650/2 indicators. All others reside in the record but are unused.

• Because filing is not needed in an online catalog, we ignore the filing indicators. We also remove tags, indicators, and subfield delimiters from our online catalog records. They are not needed and only confuse the user.

• We only store select information from each MARC record and store it in a different Internal Format.

DIACRITICS, SPECIAL CHARACTERS, ETC.

Use of the standard ALA 162-character set was much less common than standardization in other areas surveyed. When asked whether their systems presently read, stored, and output the ALA 162-character set, 67 percent of the respondents said they did while 33 percent did not. A higher number of respondents (78 percent) indicated they accepted, stored, and output diacritics and special characters. The systems of two respondents did not include both upper- and lowercase characters. Negative responses to questions about character sets elicited the following comments:

• The system accepts the full ASCII character set which includes special characters.

• The systems use a very close approximation to the ALA character set, but local needs have not brought all 162 characters into use.

• We would use the full ALA character set if available on an inexpensive terminal or microcomputer. Most users do not need the full set. Upper/lower case, however, is vital. We use the 128 char[acter] ASCII set.

• Apple IIPlus or IIe keyboard characters only. In addition no " [quotation mark] can be used in cataloging data.

• The MARC system at [vendor X] presently supports 98 EBCDIC character set. [Vendor X] now has the capability to support the 162 ALA character set, including diacritics and special characters. The MARC file will be reconfigured in fall of 1984 to take advantage of this capability.

• Translation table utilized for nonprinting characters.

• Certain characters cause unpredictable mayhem when displaying on the CRT.

EXTRA-LENGTH RECORDS

The MARC format allows specification of a record length of up to 99,999 data characters. Tape standards call for transmission of data on magnetic tapes in 2,048 character blocks. Various conventions are used to handle records exceeding 2,048 characters. Those being surveyed were asked the fairly general question, "How does your system handle extra length records?"

Of the 39 respondents to this question, 62 percent indicated there was some length limit in their systems, although it varied greatly. The comments call attention to several areas that system purchasers should investigate.

The following comments concentrated on the issue of maximum field and record sizes:

• [Vendor X] establishes a practical limit [of] 4,096. This could be any number up to 32,256. There are some technical disadvantages to extremely large maximum record lengths. If a record exceeds the number set for the system, the record is truncated.

• Internal record lengths may vary up to 8,188 characters. The largest on file is 6,000 + characters.

• Unusually long records trigger error reports.

• Our maximum length, for both storage and exchange is 6,000 bytes. Records larger than this are rejected if new or, if an existing record of <6,000 bytes is lengthened over that limit through additional fields put in during a correction/amendment process, then fields are "chopped" one by one until the limit is achieved.

• We have decided to limit the amount of data that can be printed on one card and the number of continuation cards. Our programming can handle a maximum of five card main entr[ies] (four continuation cards). When a record is longer than this, it is checked by the head librarian. Frequently, adding subfield b to the 245 tag or shortening a contents note will allow the data to fit into the five cards.

• The [Y] system can accept a MARC-formatted record up to 10,000 bytes in length (the maximum OCLC/MARC record length is 4,096 bytes). A record exceeding the 10,000-byte maximum is rejected and not processed by the system.

• System is capable of handling records 4,096 characters in length. Records which exceed this length are truncated.

• The [system Y] programs are designed to process records as long as 8,192 bytes (a de facto maximum length established by LC). Current buffers are set at only 3,200 bytes, as we have not yet encountered records any larger.

• Data fields (and full record) have maximum number of characters. No longer records possible.

Other comments concerned the tape format for longer records.

• A logical record can contain a maximum of 99,999 data characters. Extra-length records may occupy more than one physical record (2,048 bytes), but in no case may the[y] exceed the logical record limit. Records spanning more than one physical record are possible through the use of the 5-character "Segment Control Word" described in appendix III.A of the *MARC Formats for Bibliographic Data*.

• Use max[imum] blocksize of 2,048.

• It is not clear what you mean by "extra length records" but the following information may be useful. The [Y] system does not use fixed-length records, only variable-length records. Spanned records on tape are handled automatically by the system.

• Records are stored as variable blocked records with a maximum record length of 8,192. Output procedures capable of providing 2,048 increments as well as variable blocked.

• It unblocks spanned records to create a full MARC record in the system.

FUTURE ENHANCEMENTS

The final survey question was, "If your system does not meet in all respects the definition of MARC compatibility at the present time, please describe any future enhancements." Responses suggest that many system suppliers are aware of the need to support the MARC format and are moving to do so as much as possible. Plans to add full support of diacritics were mentioned by a number of respondents. Some said that development in this area was dependent on the availability of low-cost terminals and printers capable of handling the ALA character set. One vendor plans to move ahead in a newly developing area of MARC standards. This response stated, in part:

• [Vendor X] hopes especially to promote MARC-based standards for serial holdings information. No specific timetable can be given for this development but [vendor X] will respond, as in the past, to user needs in order to build the most appropriate systems for its customers.

The need for standardization may not be universally recognized, however, as seen in the following response.

• [System Y] is an integrated library system oriented to the special libraries market. We do not plan in the near future, at least, to enhance the system so that MARC formats can be stored and output since the special library market seems to have little interest in consortia or other kinds of networking which would necessitate storing MARC tags and indicators. If we decide to expand our market to include academic, public and medical libraries, we will certainly need to add this capability.

QUESTIONS FOR VENDORS

Responses to the TESLA survey show varying degrees of implementation of the MARC format. They suggest a series of questions that might be posed to systems vendors by those investigating automated library systems. Evaluation of responses to these questions should give libraries a good picture of whether the system they are considering is truly MARC compatible.

1. Does the system accept MARC records?
2. Does the system store MARC records?

3. Does the system output MARC records?

4. Are MARC formats for all types of materials (serials, maps, music, etc.) supported?

5. Are all defined fields supported?

6. Are all MARC fixed fields accepted, stored, and output?

7. Are indicators supported for all fields?

8. Are all defined subfields supported?

9. How soon are newly defined fields, subfields, and indicators available for use?

10. Does the system support the expanded American Standard Code for Information Exchange (ASCII) character set for MARC records?[3]

11. Are there any restrictions on field or record length? If so, what is done when the limit is exceeded?

12. How are records that exceed the standard 2,048-byte physical record block size handled?

CONCLUSIONS

Vendors seem to be more aware of the importance of the MARC format than they were several years ago when TESLA first became interested in the issue of MARC compatibility. However, one still cannot assume that the MARC format is always fully implemented. System suppliers should be questioned carefully. A library's future automation efforts could be hampered if the current system does not provide for the preservation and output of MARC-format bibliographic data.

REFERENCES

1. For an extended discussion and definition of MARC compatibility, see Walt Crawford, *MARC for Library Use* (White Plains, N.Y.: Knowledge Industry, 1984), p.151–67.
2. To preserve respondent anonymity, [vendor X] or [system Y] is substituted for the name of a vendor or proprietary product in all respondent quotes.
3. For specifications, see U.S. Library of Congress, Automated Systems Office, *MARC Formats for Bibliographic Data* (Washington, D.C.: Library of Congress), appendix III.B.1.

APPENDIX A. MARC COMPATIBILITY SURVEY

Definition

"The US/MARC formats are standards for the representation of bibliographic authority information in machine readable form. The structure of US/MARC records is an implementation of the American National Standard for Information Interchange on Magnetic Tape (ANSI Z39.2–1979) and of Documentation—Format for Bibliographic Information Interchange on Magnetic Tape (ISO 2709–1973)."*

US/MARC standards are accepted by librarians, networks and vendors across the nation. The term MARC compatibility means that any given system will accept as input full MARC records, store complete records with all fields (fixed and variable) including indicators and special characters in a uniquely identifiable manner, and be capable of outputting a full MARC record with no loss of information.

Survey Questions

A. *MARC Formats*

1. Will your system presently accept, store and output MARC records for the following formats:

Books	☐ Yes	☐ No
Film	☐ Yes	☐ No
Manuscripts	☐ Yes	☐ No
Maps	☐ Yes	☐ No
Music	☐ Yes	☐ No
Serials	☐ Yes	☐ No

* John Attig, "The US/MARC Formats—Underlying Principles," *Information Technology and Libraries* 1,no. 2:169–74 (June 1982).

2. If the answer to question 1 was NO for all formats, please explain the format used by your system.

B. *Fixed Fields*

1. Does your system presently accept, store, and output all fixed fields from all MARC formats?

☐ Yes ☐ No

2. If your system has any limitations regarding fixed fields and you answered NO to question 1, please explain.

☐ Yes ☐ No

C. *Variable Fields*

1. Does your system presently accept, store, and output all variable field *tags* from all MARC formats?

☐ Yes ☐ No

2. Does your system presently accept, store, and output all variable field *indicators* from all MARC formats?

☐ Yes ☐ No

3. Does your system presently accept, store, and output all variable *subfield delimiters* from all MARC formats?

☐ Yes ☐ No

4. If the answer to 1, 2 and/or 3 was NO, please explain in detail the limitations of your system. (For example: If your system does not retain indicators specifying non-filing characters and removes leading articles and non-filing indicators, please state this.)

D. *Diacritics, Special Characters, Etc.*

1. Does your system presently read, store, and output the 162 ALA character set?

☐ Yes ☐ No

2. Does your system presently accept, store, and output diacritics and special characters?

☐ Yes ☐ No

3. Does your system presently utilize both upper and lower case characters?

☐ Yes ☐ No

4. If your answer to 1, 2, and/or 3 was NO, please explain the limitations of your system.

E. *Other*

1. How does your system handle extra length records?

2. If there is any other way in which your system does not meet the definition of MARC compatible provided on page 1, please explain.

F. *Future Enhancements*

1. If your system does not meet in all aspects the definition of MARC compatibility at the present time, please describe any future enhancements which are in development. Include an estimated time frame for their availability. ■■

Special Tutorial

Data Relationships: Bibliographic Information Retrieval Systems and Database Management Systems

M. E. D. Koenig

The use of database management systems (DBMSs) to underlie library and information system automation is a development of increasing importance. Modern DBMSs are based upon the concept of "normalizing" data, that is, breaking down data records and data relationships into their most basic components and then relating those components to allow data retrieval in a variety of fashions to a variety of users. Ideally the normalization process allows the DBMS to respond appropriately to questions quite unanticipated by the system designers.

The process of relating the data elements to each other requires what might be termed data structures of the second type (hierarchical, network, and relational), as distinct from and in addition to data structures of the first type (inverted files, multithreaded files, etc.) used in typical bibliographic information retrieval systems. This article examines the second type of data structures, attempts by example to elucidate them, and explores some of the implications of DBMS development for library and information technology.

INTRODUCTION

Database management systems (DBMSs) have emerged in recent years as the foundation of numerous major library automation efforts. For example, the WLN (Washington Library Network) system is based on a DBMS called ADABAS[1], as is BLIS (Biblio-Technique Library Information System), the commercial version of WLN. Basing the WLN system on a DBMS rather than building it from scratch enabled WLN to emerge full grown, as it were, on the library scene in 1978, in contrast to OCLC and RLIN, which can trace their scratch-built lineage back a decade and a half. DBMSs are also used for complex commercial and scientific systems.[2] It is the intent of this paper to discuss the data structures on which DBMSs are based.

Typically, DBMSs were developed, starting in the 1960s, by corporations for their internal use.[3] The impetus was the replacement of sequential tape storage by direct-access disc storage, which allowed much more prompt and efficient access but was substantially more expensive per record stored. The job of the database management system was to store data nonreduntantly and make it accessible to various users for various purposes. To do this DBMSs need: a data definition language (DDL) by which the users define for the system what the data is, what its characteristics are, and on what basis it will need to be searched; a data manipulation language (DML) by which the users can retrieve the data once stored; and a data dictionary/directory (DD/D), the dictionary component being essentially a thesaurus of what the data consists of intellectually, alternative names, etc., and the directory component recording the data processing characteristics of the data, field type, arithmetic or alphabetic characteristics, etc.

The heart of the system is the data management software that, given the appropriate

---

M. E. D. Koenig is a faculty member of the School of Library Service, Columbia University, New York.

information, can store the data and retrieve it again. The basic functions of DBMS are to organize data, to provide access facilities, and to provide control facilities.

The essence of a DBMS is that they are designed to respond to various users who have various and differing needs. This makes a DBMS quite distinct from bibliographic information retrieval systems, which basically have only one purpose—to select and retrieve from a large set of very similarly structured records. Those records exist quite independent of and unrelated to each other (as far as the data file is concerned). A typical bibliographic information retrieval system uses what might be called data structures of the first type, such as inverted files (which are simply indexes in their essence) or multithreaded files (which accomplish the same function) to locate the appropriate records.

An inverted file, as it is termed by data processors, is simply an index file. The terminology came about in the early days of data processing, when large data files had to be stored on tape because direct access storage was much too expensive for large files. The searching of such files had to be sequential, that is, the records were searched through sequentially to get to the data within the records. This is analogous to searching for data in a library by reading through all the books in the library in accession number order, methodical but crude. When high-capacity disc storage was developed, data processors rediscovered the index. An index to a file could be mounted on disc, the disc searched just as a card catalog is searched, and the item then directly retrieved. To the data-processing community, however, this was an inversion of what they had come to regard as the natural order. With an index, one was using pieces of the data (the entry points in the index) to get at the record rather than searching through the records to get at the data, hence they called their new discovery—their index—an inverted file. In actual application, an inverted file system becomes rather complex, with several layers of indexes to the indexes to the indexes.

A multithreaded file is most easily described as an electronic equivalent of a library in which the entry (author's name or subject heading, for example) in the catalog simply directs one to the first book in the stack (the stacks being open access, but shelved in accession number order) on that subject. At the end of the first book on, let's say, U. S. Grant, would be a "pointer" to the accession number of the second book on U. S. Grant, and at the end of that book, a pointer to the third, etc. When a new book is added on a subject already in the collection, the catalog remains unchanged, but a new pointer is added for the new addition. One would retrieve all the books on U. S. Grant by tracing the thread (or chain or link as it is also called) through the stacks. Since a book can be on multiple subjects, it can also be on multiple threads, hence the terminology multithreading, multichaining, multilink, etc. To search for a book on U.S. Civil War history and on economics one would select the shorter thread, trace it through, and retrieve all items that also had a pointer for the other thread. In a library environment the scheme does not sound very plausible. In a data-processing, direct-access environment, however, the system works tolerably well. Access tends not to be quite as prompt as in an inverted file environment, but overhead and maintenance of the indexes is far less troublesome.

DBMS, however, relate the different records to each other, and this requires what might be termed data structures of the second type. These data structures are not analogous to traditional library practices, and it is the intent of this article to describe them in the most straightforward method, which is to present specific examples and compare DBMS data handling to conventional bibliographic information retrieval systems.

CONVENTIONAL BIBLIOGRAPHIC INFORMATION RETRIEVAL SYSTEMS

The concept of a typical bibliographic information retrieval system is that of a data file of similar and quite independent records. Typically, these records are the descriptions of individual bibliographic items. A simplified example of such a record is shown in figure 1. Such a file would commonly be mounted online with inverted indexes (analogous in their function to a card catalog) of those descriptors or entry points on which it has been deter-

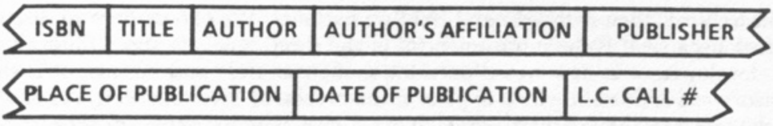


Fig. 1. Simplified Bibliographic Record.

mined to make the record quickly accessible (that is accessible through the indexes rather than simply by brute force sequential searching through the file). Those descriptors or entry points could correspond to fields within the record such as, in our example, author or date, or they might represent subfields or components of a field, such as subfields of place (city, state, country) of publication or components such as the substantive words of the title. Such a system with the appropriate indexes would readily respond to a search request for all articles published in *MIS Quarterly* with the words "information" and "requirements" in their title. That is primarily what a bibliographic information retrieval system is designed to do—locate a set of records that match certain search requirements, with those records in turn corresponding to particular bibliographic items. Consider, however, how we might make a relatively simple request, such as supplying the names and addresses of all publishers in New York City, Philadelphia, or Boston who have published engineering books in the last five years, of the same system. All of the necessary data are in the file—publishers, their addresses, a call number that would identify engineering books, and their date of publication. The closest the system could come to answering the request would be to identify and produce all of the records identifying engineering books published in New York, Philadelphia, or Boston in the last five years. If we printed that list, it would contain the information we wanted, but instead of a list of publishers, we would have a much longer list of books, and McGraw-Hill, instead of appearing once, would appear as many times as there were McGraw-Hill engineering books published within the last five years in the database.

The conventional bibliographic information retrieval system has been designed, implicitly at least, to incorporate only a modest ability to relate data. The data relationships are based on the concept of relating terms and descriptions to the records of individual bibliographic items, and there is effectively no linkage between those records other than the sharing of common elements or descritions. The database structure is designed to relate only to bibliographic items. It is not designed to relate, for example, to publishers. There is an implicit assumption that all users will want to use the file in fairly similar fashion—to obtain access to records describing bibliographic items.

THE DBMS ENVIRONMENT

The DBMS environment, by contrast, is designed on the assumption that various users will need access to various files for various purposes, and that those files need to be related to each other because a user's information needs may require accessing several files in answering just one query. This multiplicity of files and applications is so basic to the concept of database management that those persons grounded in the DBMS context are loath even to refer to bibliographic information retrieval systems as databases. The DIALOG system, to a DBMS person, is simply a collection of data files; it is not a database nor even a collection of databases. What is missing is the multiplicity of applications and any significant degree of interrelationship of the files. Chemical Abstracts or Medline or Scisearch files are massive indeed, but they are, in the eyes of the DBMS community, files not databases.

There are three conventional ways of relating data files to construct a database: hierarchical, network, and relational. The most recent development, and in many ways the most elegant concept, is that of the relational model. Conventionally, when discussing DBMS data structures one discusses those structures in their order of chronological appearance:

first hierarchical, then network, and finally relational. We choose here to discuss relational first because it is the most powerful of the three concepts and almost all current DBMS development is now based on relational structures, and because it is the most straightforward context in which to present the process of normalization, the decomposition of data relationships to their most elemental level, which is a concept of central importance.

RELATIONAL DATA STRUCTURES

The basic building block of a relational database is the notion of a flat table, a two dimensional, "square" (or, more accurately, rectangular) array of data in which the rows represent entities, and the columns represent attributes. Our previous example might be thought of as such a flat table, presented in figure 2.

ATTRIBUTES

	ISBN	TITLE	AUTHOR	AFFILIATION	PUBLISHER	PLACE	DATE	CALL #
Entities:	"	"	"	"	"	"	"	"
Bibliographic Items	"	"	"	"	"	"	"	"

Fig 2. Tabular Representation of a Simple Bibliographic Record.

From the point of view of bibliographic information retrieval systems, the table above is an adequate way of thinking about the data. From the database viewpoint, however, particularly in an environment where the data may be frequently modified, there are a number of problems with this table. One problem is that of space utilization and redundancy of data. ISBNs are unique, but McGraw-Hill and its place of publication will occur repeatedly. Not only does this waste space (and the efficient use of storage space was a major factor in DBMS design), but it also permits inconsistency. While a bibliographic "database" may be relatively static—new records or entities may be added but existing ones are seldom changed—such is frequently not the case in other applications (financial systems, production and inventory systems, etc.), which are much more volatile, and ease and consistency of updating becomes much more important.

The same is true for author and affiliation; Columbia University will probably appear repeatedly in a file of any size. We can attack those problems by breaking the flat table above down into a number of smaller and more basic tables, as shown in figure 3.

ATTRIBUTES

FILE 1	ISBN	TITLE	AUTHOR	PUBLISHER	PLACE	DATE	CALL #
Entities							

ATTRIBUTES

FILE 2	AUTHOR	AFFILIATION
Entities		

ATTRIBUTES

FILE 3	PUBLISHER	PLACE
Entities		

Fig. 3. Decomposition (Normalization) into Smaller Tables.

We now have more tables, but we have saved space. We need only record once that McGraw-Hill's place of publication is New York. We have also achieved a number of other benefits. Perhaps the most obvious at this stage is that we now have an easy way, via file 3, of asking that question about publishers in New York City, Philadelphia, or Boston. We now have a file in which the publisher is an entity and not just a description in a file in which the bibliographic record is the entity. That is perhaps a bit clearer if we relabel the files above, as in figure 4.

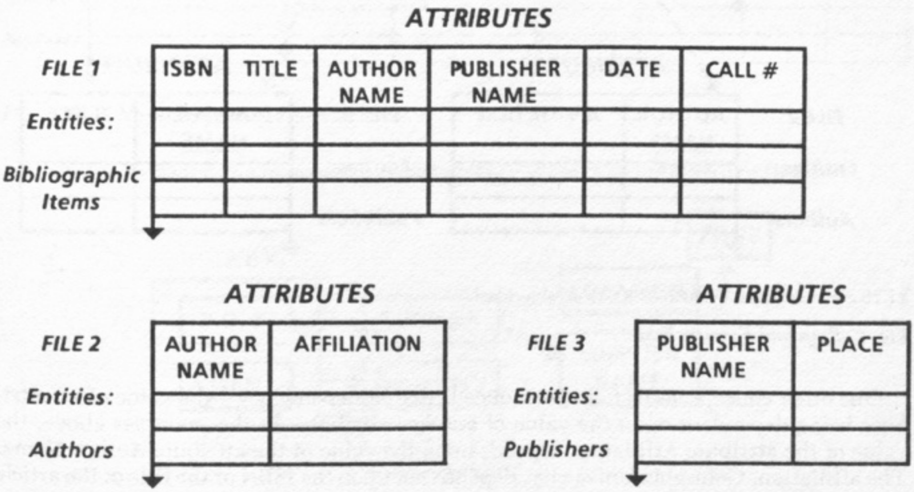


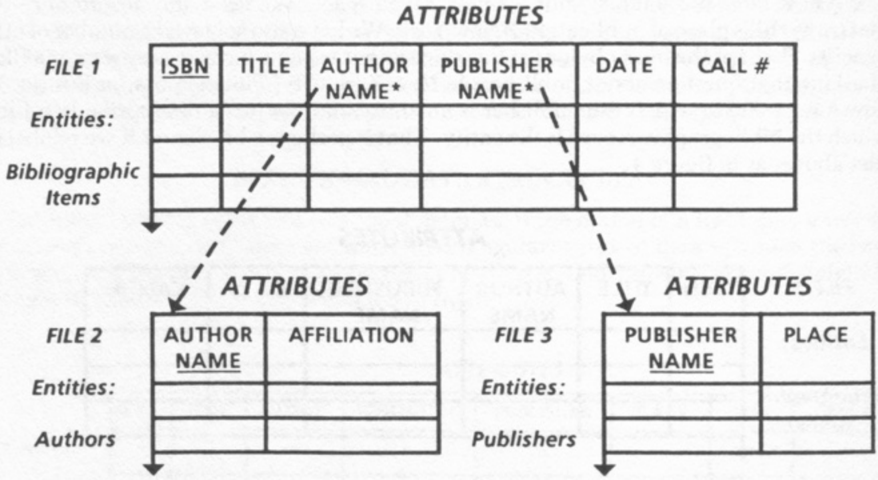
Fig. 4. Entities and Attributes.

Now it is clearer that each file represents an entity, that entities have attributes, and that common attributes like Publisher Name or Author Name can link different files.

The process that we have begun to carry out on this file is called normalization. We have normalized the tables in figures 1 and 2, which are in "first normal form" to the tables in figures 3 and 4, which are in "third normal form." This process of normalization is essentially that of reducing data relationships to their lowest common denominators—reducing the relationship to a number of atoms, the most basic components or entities of which the larger structure is composed.

To pursue the idea of normalization we need to introduce additional concepts and terms. Each row, or tuple as it is sometimes called, in a flat table is, or should be, unique and uniquely identifiable. The attribute or combination of attributes that provides the unique identification in each table or relationship is that table's or relationship's key. (Note that this use of "key" is not the same as the concept of search key that is used in bibliographic information retrieval systems. The key in a relational sense must also be a search key in the bibliographic sense, as that key is the one way of being sure of getting to that record. Other attributes in a relational structure may be made searchable, that is keys in the bibliographic sense, even though they are not keys in the relationship sense.) An attribute that is a key in one table and a link to another table is called a foreign key in that other table. Identifying the keys in figure 4 results in figure 5.

In figure 5 and throughout the remainder of this paper, Author Name and Publisher Name are the keys to files 2 and 3, respectively, and are foreign keys in file 1, for which ISBN is the key. Call Number is a candidate key for file 1, it is also capable (assuming it is a unique call number, not merely a class number) of uniquely identifying the entities in file 1. Call Number and other attributes, such as Title, can of course be made search keys in the bibliographic sense even though they are not keys in the relational sense.



KEYS are underlined, FOREIGN KEYS are asterisked.

Fig. 5. Keys and Foreign Keys.

One other concept needs to be introduced, that of dependency—the value of an attribute being dependent upon the value of another attribute. In the examples above, the value of the attribute Affiliation depends upon the value of the attribute Author Name. The affiliation, Columbia University, depends not upon the ISBN or the title of the article or the journal name but upon who the author is.

Another way of describing what we have done in normalizing figure 2 to figures 3, 4, and 5 is to say that we have reduced the relationship of figure 2 to simpler relationships in which the value attached to each attribute is a direct function of or is dependent upon the value of the key attribute. We could diagram those dependencies as shown in figure 6.

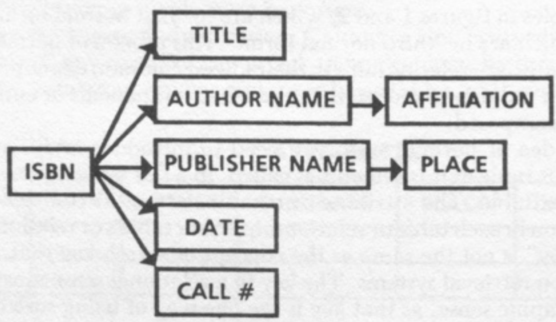


Fig. 6. Dependency Relationship.

The dependence of Affiliation upon ISBN, for example, is transitive; it is not dependent upon ISBN directly, but its dependence is via Author Name, upon which Affiliation is directly dependent and which in turn is directly dependent upon ISBN.

There is another form of dependency called partial dependency and to illustrate this we will hypothesize another situation, let's say a purchase order file in a library. The entity here is the purchase, which is uniquely identifiable by the combination of ISBN and Supplier (remember that a key can be composed of more than one attribute). We order numerous ISBNs from the same supplier, and, particularly with branch libraries, we not infrequently order the same ISBN from different suppliers (that is a many-to-many relationship, a point we shall return to later). From the acquisitions librarian's point of view, the data need to be looked at on the basis of an entry on a particular purchase order and should look something like figure 7.

	P.O. #	ISBN	SUPPLIER #	SUPPLIER NAME	LOCATION	QUANTITY	PRICE	ZONE
Entities:								
Purchases								

Fig. 7. First Normal Form of Purchase Data.

If we try to sketch the dependency relationship, it looks something like figure 8.

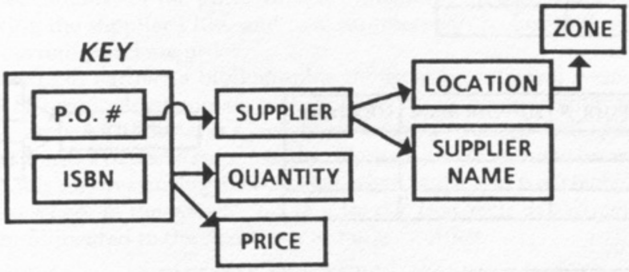


Fig. 8. Purchase Data Dependency Relationship.

Quantity and Price are dependent upon the key, the combination of ISBN and Supplier Number but Supplier Name and Location are dependent upon only a part of the key, being dependent upon Supplier Number, which is in turn dependent upon Purchase Order Number, regardless of ISBN. That is, Supplier Number is partially dependent, while Location and Supplier Name are dependent upon Supplier Numbers and Zone is in turn dependent upon Location. These relationships are realized from our own knowledge, they are not inherent in the as yet unnormalized data structure.

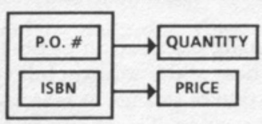
Figure 7 is in first normal form, a simple flat table. If we remove the partial dependencies, we arrive at second normal form, as shown in figure 9.

ATTRIBUTES

FILE 1	P.O. #	ISBN	QUANTITY	PRICE
Entities:				
Purchases				

FILE 2	P.O. #	SUPPLIER #	SUPPLIER NAME	LOCATION	ZONE
Entities:					
P.O. Supplier					

DEPENDENCIES



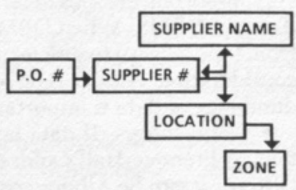


Fig. 9. Second Normal (Partial Dependencies Removed).

We still have the transitive dependency of Tax being dependent upon Location, which is in turn dependent upon Supplier Number. If we remove all transitive dependencies as well as partial dependencies, we arrive, as we did in the first example, at the third normal form, displayed in figure 10, where the data relationship and dependencies are explicit in the data structure.

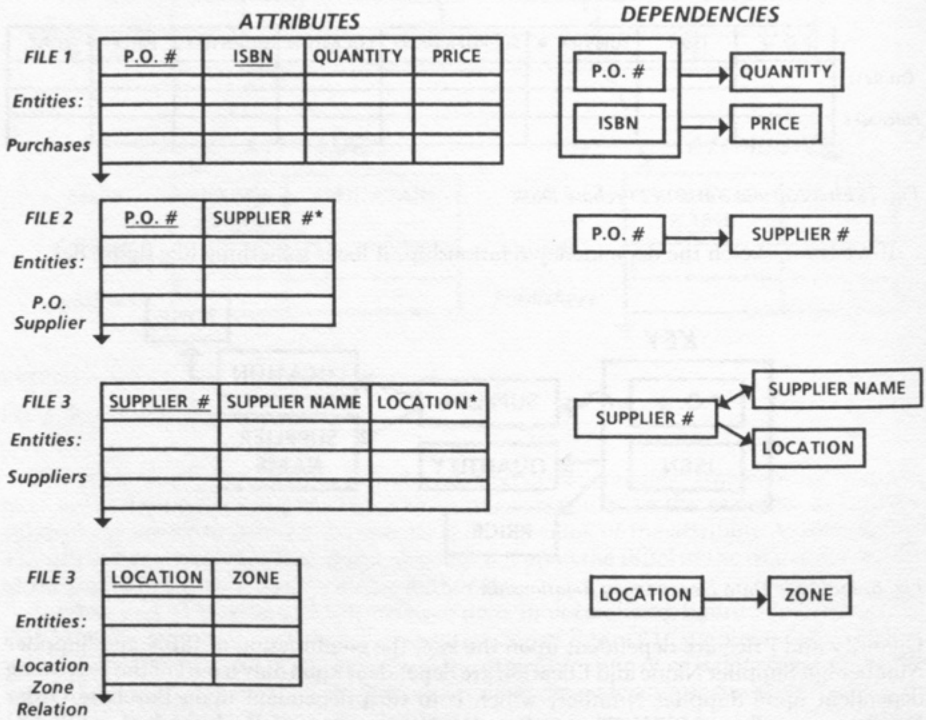


Fig. 10. Third Normal Form (Partial and Transitive Dependencies Removed).

For most practical purposes, the third normal form represents the decompostion or normalization of data relationships to their most basic level, their lowest common denominators. There are, in fact, somewhat higher levels of normality, for example fourth normal form and Boyce-Codd normal form,[4] but the distinction between third normal form and higher forms is esoteric, functionally not very significant, and beyond the scope of this paper.

We have already discussed in passing the ability to describe relationships beyond the complexity of those addressable in bibliographic information retrieval systems and more efficient utilization of storage as reasons for normalizing data. These are in fact the two very important reasons. The ability to address complex relationships of data is the raison d'etre of DBMS. When DBMS were first being developed, and data storage was moving from tape (cheap) to disk (expensive), the efficient utilization of storage space was a major consideration. Now it is no longer so important economically, but the concept of nonredundancy of data is important for other reasons as well:

- Consistency—If data is nonredundant (only in the file once), it cannot be inconsistent. If Prentice-Hall's address is only in the file once (figures 3–10) it can be Englewood Cliffs or it can be Albuquerque but it cannot be both as it could be in figure 2.
- Updating, Convience, and Correctness—If Harcourt Brace Jovanovich moves from

New York to Florida, that information need only be modified in one place, not in the record for every book published by HBJ. Furthermore, if there is only one place to make the change, not only is there less effort for the system, but we cannot overlook other places where the change should be made as we might easily do in figure 2. This ensures consistency.

• Delete and Insert Anomalies—By breaking our data relationships down into basic entities we avoid what are called delete or insert anomalies. For example, consider the purchase order file in figure 7. We have data on a supplier in that file only as long as we have an active purchase order in that file relating to that supplier. If we delete the last purchase order in the file relating to Bowker because the transaction has been completed before entering another item relating to Bowker, we have, as far as the database is concerned, lost all information relating to Bowker, and if another order is now to be created, all the Bowker data must be re-created. That is the delete anomaly. The insert anomaly is basically the same concept. We cannot add or insert data on a supplier unless there is a purchase order to that supplier or unless we create some agreed upon conventions for a dummy or false order. By contrast, after normalization (see figure 10) the supplier entities exist quite independently of the purchase order entities. Purchase orders can be deleted without affecting the supplier's file, and new suppliers can be inserted regardless of the existence of a current purchase order.

As was pointed out earlier, a bibliographic information retrieval system is relatively static (or in D.P. terminology nonvolatile), but in a more dynamic environment the advantages above are very important. Consider a database management system of student records. If you submit a paper to complete the requirements of a course for which you have an incomplete, it is very reassuring to know that when that grade is corrected or updated it is updated everywhere in the system. There is no risk that your department's correction will not be communicated to the registrar's or bursar's office.

SCHEMAS AND SUBSCHEMAS

The examples we have used are relatively simple ones that only begin to illustrate the complexity of relationships that must be made explicit in a typical DBMS environment. Let us think, for example, of the relationships that must exist in a DBMS that represents the acquisitions and processing efforts of a major library. How would we design and set up such a system? Clearly, we would want to think of how the system would and should be used, by whom, and for what purposes. We could begin by talking with users and potential users about how they would and could use such a system, what information they would need to have access to and for what purposes, how they would need to retrieve it, etc. We would start by assembling what are called user views. We might think of figures 6–10 as representing the normalization of one component of the user view of an acquisitions librarian. That person might have said, "There are times when for such and such a reason I need to enter the system by ISBN and Supplier Number and be able to see the following data: supplier name, supplier location, quantity of order, price, and tax."

In figures 6–10 we have taken that user view and normalized it. That person, however, presumably would have needed to do other things as well (and we as analysts may well identify further things that they could or should do on the system, but that they have not thought of because they do not now do it that way and have not imagined the system to be capable of supporting such an operation). If we combine all the things that a person wants or needs to do with the system, we have what is called a subschema, for example, the subschema of monograph ordering. If we combine the subschemas of all the various users and potential users of the systems, we arrive at the overall schema or integrated data view of the DBMS.

To illustrate that process of combination or integration, let us suppose that the same user has also said, "Another thing I need to be able to do is to look at acquisitions by what fund the money comes from and be able to look at each fund or source of money and see what we have bought, how much we have spent, from what supplier and where that supplier is located, as some funds have stipulations on where and how the money is expended." If we tried to sketch the dependency relationships in that statement we would come up with the relationship set out in figure 11.

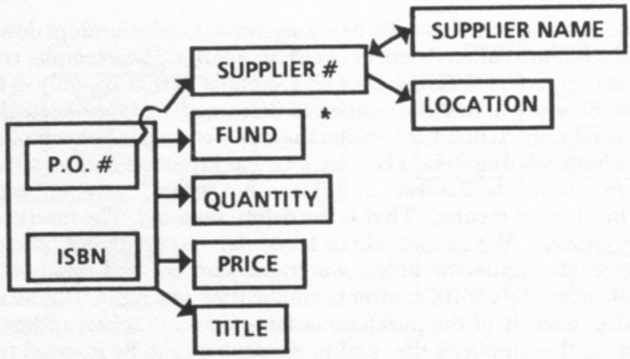


Fig. 11. Second User View, Dependency Relationship.

Reducing that relationship to third normal form, as we did before in figures 6–10, we come up with the tables in figure 12.

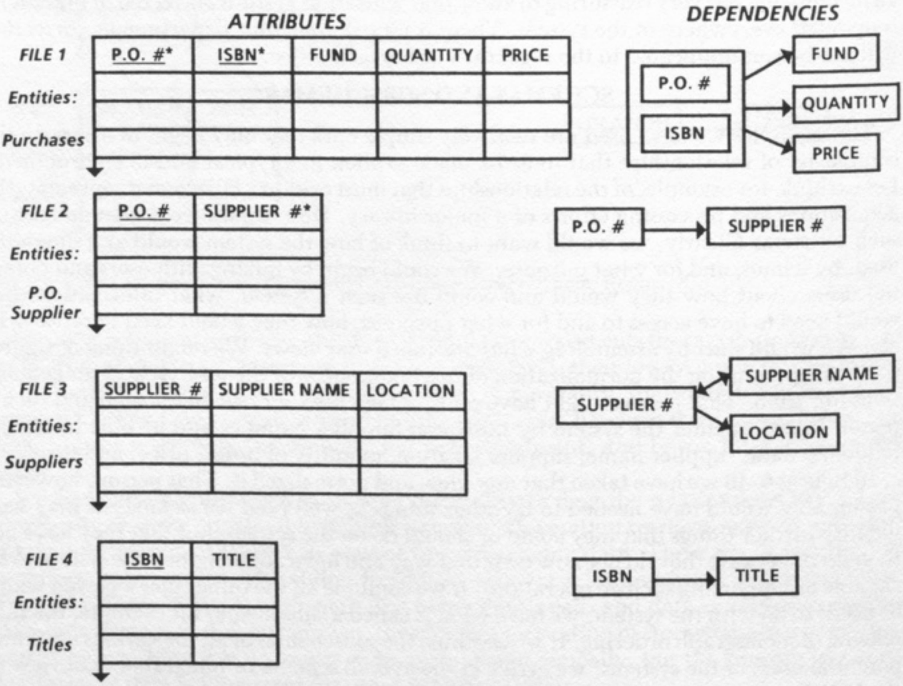


Fig. 12. Second User View, Normalized.

By comparing figures 10 and 12 it is apparent that the two user views are not unrelated. Two of the files are identical in their essentials, the Suppliers files (file 3 in figure 10, and file 3 in figure 12), and the P.O. Suppliers files (file 2 in figure 10, and file 2 in figure 12).

The only differences are in the linking of attributes to other files (the functioning of attributes as foreign keys). By combining files 2, 3, and 4 of figure 10, with files 1 and 4 of figure 12 we can combine both user views into one subschema. Now we have a dependency relationship that looks like figure 13.

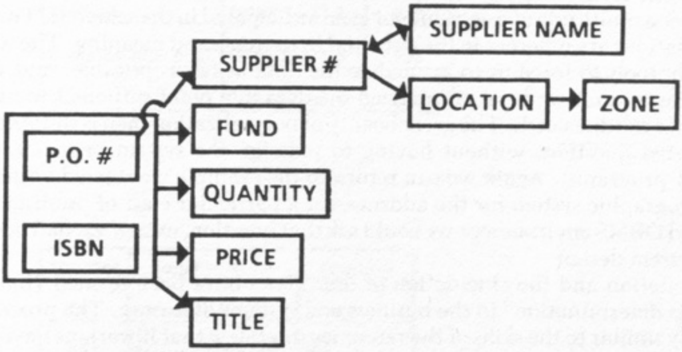


Fig. 13. Combined Dependency Relationship (User Views 1 and 2).

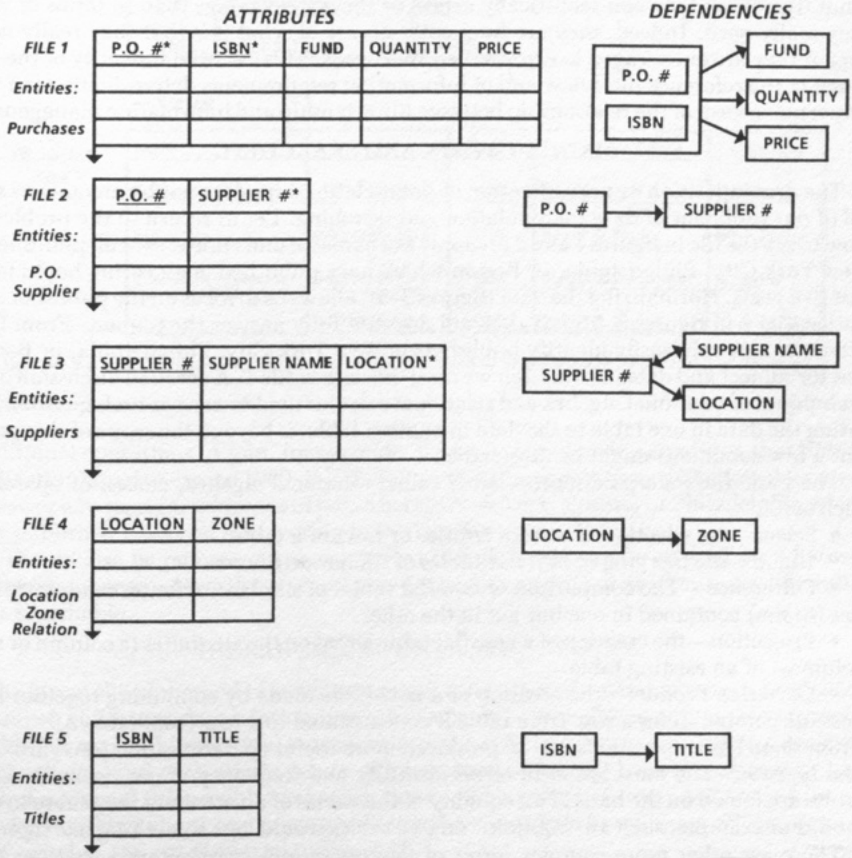


Fig. 14. Subschema (User Views 1 and 2 Combined and Normalized).

Just as we can assemble our hypothetical user's views into a subschema, we can assemble subschemas into an overall schema for the database management system.

The ultimate benefit, however, of normalizing data and amalgamating subschemas into schemas is that if we can normalize the data to its least common denominators, its most atomistic form, and establish the relations thus determined, then we can theoretically answer a multitude of questions not even anticipated in the schemas. Thus we can say that the relations are inherent in the data and in its usage and meaning. The user views are then simply tools to force us to normalize the data as far as possible, and they serve as checks to insure that we have normalized the data to a point sufficient to handle the organizations's routine needs. The great beauty of normalization then is the ability to handle unanticipated questions without having to redesign the system or generate new code (write new programs). Again we can return to the example we started with: How do we ask a bibliographic system for the addresses of a particular class of publishers? In a well normalized DBMS environment we could ask that question, even if we had not anticipated it in the system design.

The elicitation and the elucidation of user views have been termed "information requirements determination" in the business and systems literature. The process, however, is strikingly similar to the skills of the reference interview that librarians have been honing for decades. How do you find out what information it is that the user really needs, particularly when the users typically have very imperfect notions of what the information system, whether library or DBMS, can really provide and tend to phrase their requests in terms of what they think they can realistically expect of the system rather than in terms of what they really need. Indeed, they are frequently unsure of what it is that they really need, even if they are encouraged to express their real needs. This essential indentity of the processes of the reference interview and of information requirements determination is a very important aspect of the relationship between librarianship and information management.

MANIPULATION AND SEARCHING

The decomposition or normalization of data relationships does not by any means solve all of our problems of data manipulation and searching. Let us return to the problem of how to ask the file in figures 1 and 2 to supply the names of and addresses of all publishers in New York City, Philadelphia, or Boston which have published engineering books in the last five years. Normalizing the data (figures 3–5), allows us to focus on the publisher as an entity (file 3 in figures 3–5), but that still does not fully answer the request. From file 3 alone we can only easily identify publishers in New York City, Philadelphia, or Boston, but for subject and date information we need the data in file 1. A detailed discussion of the techniques of relational algebra and relational calculus used to answer such questions, relating the data in one table to the data in another table, is beyond the scope of this paper, but a few comments might be illustrative.

The basic file manipulations, what is called relational algebra, consist of operations such as:

- Select—the selection of certain entities or rows of a table.
- Union—the merging of two flat tables of similar attributes.
- Difference—The comparison of two flat tables of similar attributes to identify entities (tuples) contained in one but not in the other.
- Projection—the creation of a new flat table keyed on the attributes (a column or set of columns) of an existing table.
- Cartesian Product—the creation of a new table made by combining together in all possible combinations a row from table B concatenated to a row from table A.

From these basic operations can be produced more useful operations that join various tables together. The most basic, in terms of utility and frequency of use, is one in which tables are joined on the basis of an equality of the values of an attribute they share in common. For example, such an "equijoin" on two tables would look something like figure 15.

There are other more complex forms of join operations, particulary variations upon

AN EQUIJOIN OF

FILE 1

Fund	*ISBN*	*Supplier*
E. Mason Memorial	0-07-054484-0	006
Sterling	0-914236-31-8	010
E. Mason Memorial	0-201-14471-9	008
Slush	0-914236-50-4	010

AND

FILE 2

Supplier	*Supplier Name*	*Location*
006	Blackwell N.A.	Oregon
007	Stevens & Brown	U.K.
008	Swets & Zeitlinger	Holland
009	Harrasowitz	D.B.R.
010	Baker & Taylor	New Jersey

YIELDS

PRODUCT FILE

Fund	*ISBN*	*Supplier*	*Supplier Name*	*Location*
E. Mason Memorial	0-07-054484-0	006	Blackwell, N.A.	Oregon
Sterling	0-914236-31-8	010	Baker & Taylor	New Jersey
E. Mason Memorial	0-201-14471-9	008	Swets & Zeitlinger	Holland
Slush	0-914236-50-4	010	Baker & Taylor	New Jersey

Note that file 2 is file 3 of figure 14, and file 1 is in fact the result of an equijoin of files 1 and 2 of figure 14.

Fig. 15. Example of an Equijoin.

conditional operations. A join, for example, might be based on the condition that date of publication equal or exceed 1980. It is these join and select operations that allow the system to answer a request for addresses of publishers in New York, Boston, or Philadelphia which have published works on engineering in the last five years.

It should also be pointed out that inverted files (indexes) can be created for whatever attributes are deemed worthy of searching. An attribute need not be a key in a relationship to be searchable.

HIERARCHICAL DATA STRUCTURES

The hierarchical method of structuring data is based on the commonsense observation that much of the time data relationships are of a parent-child-sibling nature. For example, a publisher may publish several journal titles, each title may have many issues, each issue many articles, and each article one or more authors. A journal issue therefore may be a child of a journal title, a sibling of other issues, and a parent of many articles. So far we have been talking about one to many relationships, one title to many issues, one issue to many articles, etc., as displayed in figure 16.

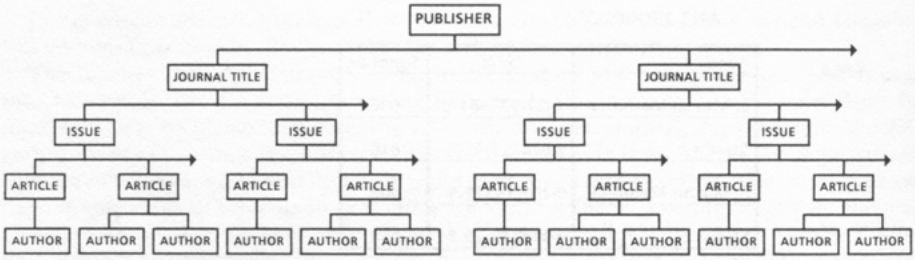


Fig. 16. Hierarchical Data Structure.

The major drawback to a hierarchical structure is revealed when we observe that there may be many-to-many data relationships. For example one article may have several authors, but one author may have produced several articles.

As we observed with figure 12 (ISBN versus Supplier), many-to-many data relationships are easily handled in a relational format, indeed they hardly even needed comment when we observed them. However, in a strict hierarchical structure a many child/many parent set of relationships is not representable. There are, however, three basic methods to get around this difficulty.

The most obvious way to get around the many-to-many problems in a hierarchical environment is to add redundant files. For example, what if we were trying to represent a situation in which a manager would typically be in charge of one or more projects and one or more employees? That could be easily represented hierarchically, as sketched in figure 17.

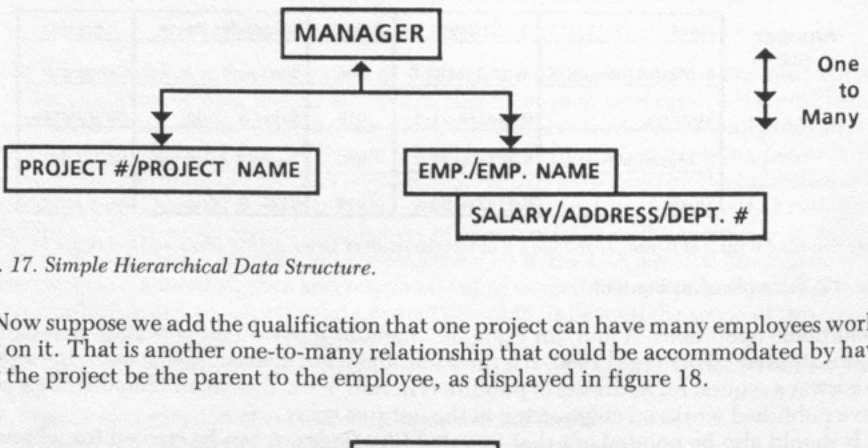


Fig. 17. Simple Hierarchical Data Structure.

Now suppose we add the qualification that one project can have many employees working on it. That is another one-to-many relationship that could be accommodated by having the project be the parent to the employee, as displayed in figure 18.

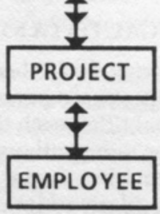


Fig. 18. Schematic of Modified Simple Hierarchical Data Structure.

But what if we wanted to be able to assign an employee to various projects simultaneously? We could add a redundant file as in figure 19.

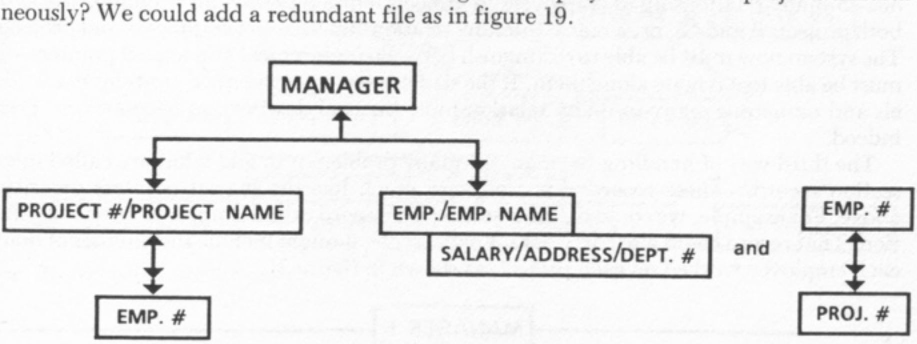


Fig. 19. "Redundant" Hierarchies.

The relationships are now represented, but we had to use two hierarchies to do it. One price that we have paid for this is redundancy. The Employee Number now appears in three places. That of course courts inconsistency. Another price is that we now must provide a mechanism for the DBMS to keep track of and interrelate the various hierarchies.

A second method is to add what are called "logical pointers" to a hierarchy. It should be added parenthetically at this point that the data storage does not physically resemble a hierarchical tree structure. The hierarchy is represented in storage by means of pointers. For example, the "manager" record will contain pointers to the addresses or locations in storage where the data representing the projects to which that manager is parent are stored. The various projects do not have to be stored adjacent to each other, as there can easily be a separate pointer for each project, or each project could have a pointer to the next project. (The parent could have a pointer to sibling 1, sibling 1 a pointer to sibling 2, etc.)

The idea of adding logical pointers is to cope with logical relationships that are typically the consequence of many-to-many situations, which cannot be directly accommodated in the hierarchical structure. The use of such pointers is illustrated in figure 20.

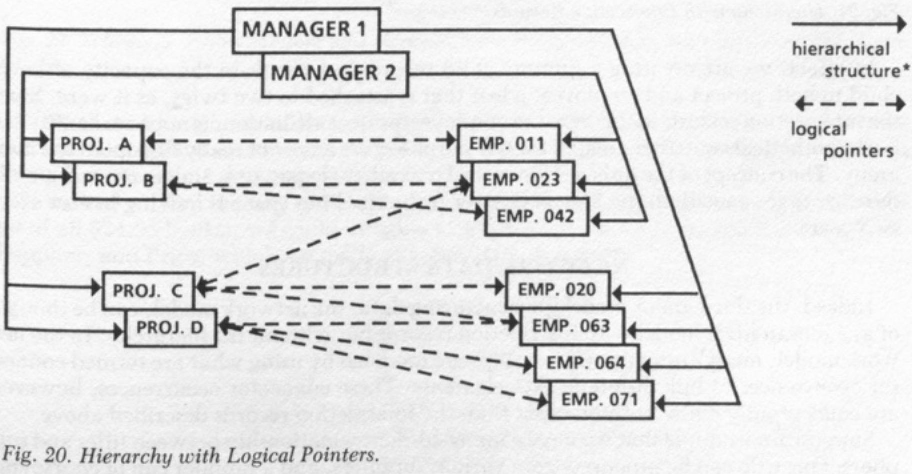


Fig. 20. Hierarchy with Logical Pointers.

In this example, in addition to a one-to-many relationship between projects and employees (project D, for example, engages employees 063, 064, and 071), we can represent a one-to-many relationship of employees to projects (employee 023, for example, works on both projects B and C), or, a many-to-many relationship between employees and projects. The system now must be able to distinguish between hierarchical and logical pointers and must be able to navigate along them. If the structure to be represented contains many levels and numerous many-to-many relationships, the complexities can become very great indeed.

The third way of handling the many-to-many problem is to add what are called intersection records. These records function very much like the logical pointers described above. For example, we could define a new type of record called Employee Project Affiliation. That record could also carry data. For example, it might include the number of hours each employee worked on each project, as shown in figure 21.

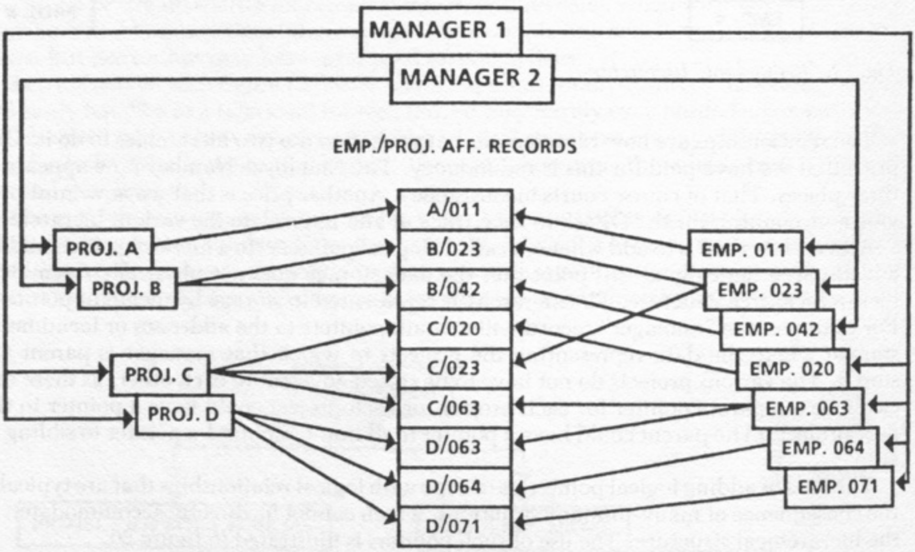


Fig. 21. Hierarchy with Intersection Records.

In effect, we are creating a dummy child record that stands in the capacity of being child to both project and employee, a leaf that is attached to two twigs, as it were. Since the intersection record, in this case the employee/project affiliation, is not a real entity but is a hypothetical construct created for our purposes, we have not really disrupted the hierarchy. The concept of the intersection record is a rather elegant one. It does use marginally more storage space than the logical pointer technique, but space is halving in cost every two years.

NETWORK DATA STRUCTURES

Indeed, the third major model for structuring data, the network model, can be thought of as a hierarchical model with intersection records but without the hierarchy. In the network model, many-to-many relationships are modeled by using what are termed connector occurrences to link different data elements. These connector occurrences, however, are conceptually really nothing other than the intersection records described above.

Suppose for example that we have a many-to-many relationship between titles and suppliers. One title can be procured from various suppliers, and a supplier can of course supply numerous books. We can link those entities in a many-to-many relationship by creating connector occurrences. Those connector occurrences, just like intersection records, may in fact carry additional information. In the case illustrated in figure 22 the connector occurrence records carry price data.

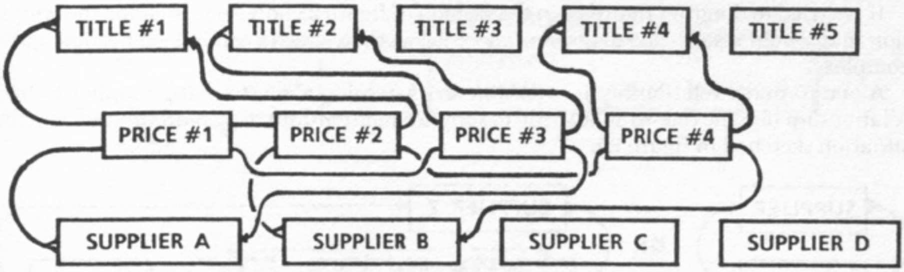


Fig. 22. Network Data Structure.

Title #1 is supplied by both Supplier A and Supplier B, for example, while Supplier B supplies both Title #1 and Title #4. The lines on figure 22 represent what are called chains or threads, which connect the entity (title or supplier in this case) to the appropriate connector occurrences. These chains are implemented by pointers, as discussed above under hierarchical structures. The Supplier A record contains a pointer to Price #1, this contains a pointer to Price #3, and this contains a pointer back to Supplier A. (Note that the pointers must not only point, they must also carry information identifying which chain they belong to, as we must be able to distinguish in Price #1 the Supplier A pointer pointing to Price #3 from the Title #1 pointer pointing to Price #2.)

We can simply and schematically represent these networks, as shown in figure 23.

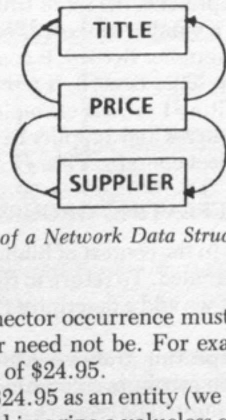


Fig. 23. Schematic Representation of a Network Data Structure (with Price as Connector Occurrence).

Note that although each connector occurrence must be unique and unambiguous, the data carried with the connector need not be. For example, both Price #1 and Price #4 might have the same data value of $24.95.

If we had wanted to think of $24.95 as an entity (we might perhaps want to think of the set of all $24.95 books), we could imagine a valueless connector occurrence linking Title, Supplier, and Price, which we might diagram as in figure 24.

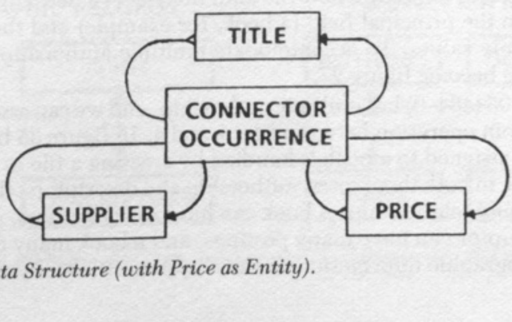


Fig. 24. Network Data Structure (with Price as Entity).

If we tried to diagram figure 24 in the fashion of figure 22 however, imagine the confusion that would result. The diagramming of network structures can indeed become quite complex.

A one-to-many relationship is even simpler. Assuming a one-to-many supplier to title relationship (and no one-to-many title to supplier relationship) we would simply have the situation sketched in figure 25.

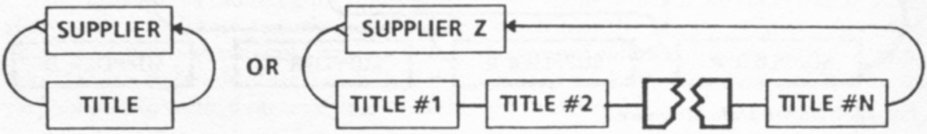


Fig. 25. One-to-Many Network Structure.

This degenerate network is simply a hierarchical one-to-many relationship, as in figure 26.

Fig. 26. Alternate Schematic of a One-to-Many Network Structure.

In database nomenclature, the relationship between entitles in a network model is generally referred to as a set. The lines we have been drawing in figures 23 and 24 above each represent a set. Thus, in figure 25 above, the line Supplier through Title #1, Title #2, and Title #N and back to Supplier Z represents the set of titles handled by Supplier Z. In the database system, the set consists of a number of records connected by pointers. Generally, one type of record, in this case the Supplier Record, is considered owner of the set, and the other type of record, in this case the Title Record, is referred to as being a member of the set. To go back to figure 22, the Title #1 set, for example, contains Price #1 and Price #2 and thereby in effect contains Supplier A and Supplier B. The Supplier B set contains Price #2 and Price #4, and thereby in effect contains Title #1 and Title #4.

REPEATING GROUPS

A question that inevitably arises in the context of bibliographic information handling is that of how repeating groups are handled. To return to figure 1, for example, what if there is more than one author, or what if we add a descriptor field and want to be able to assign multiple descriptors?

In a relational data structure, repeating groups per se are not allowed. A table must be "square," and to achieve that, each attribute column must have one and only one value (though that value may be zero or null). The reason for that insistence is to make the various file manipulations (joins, etc.) reasonably tractable. Therefore, in a network data structure the problem of repeating groups or, as it might more accurately be termed, multiple values, is handled by creating a new file and, therefore, a new entity to represent the relationship between the principal field (a book, for example) and the subfield (author) that can have multiple values. To accommodate multiple authorship, we could further normalize figure 4 to become figure 27.

Item (ISBN 0–07–054484–0) has multiple authorship, and we can associate both authors with the item by a join operation between files 1 and 4. In figure 35 below, the multiple value of descriptors assigned to a book is handled by creating a file in which the entity is Aboutness. Note that in both those cases, authorship and descriptors, the problem is really one of a many-to-many relationship. A book can have many authors, an author can have many books, a descriptor can have many postings, and a book many descriptors.

In a typical bibliographic information retrieval system, the many-to-many problem is

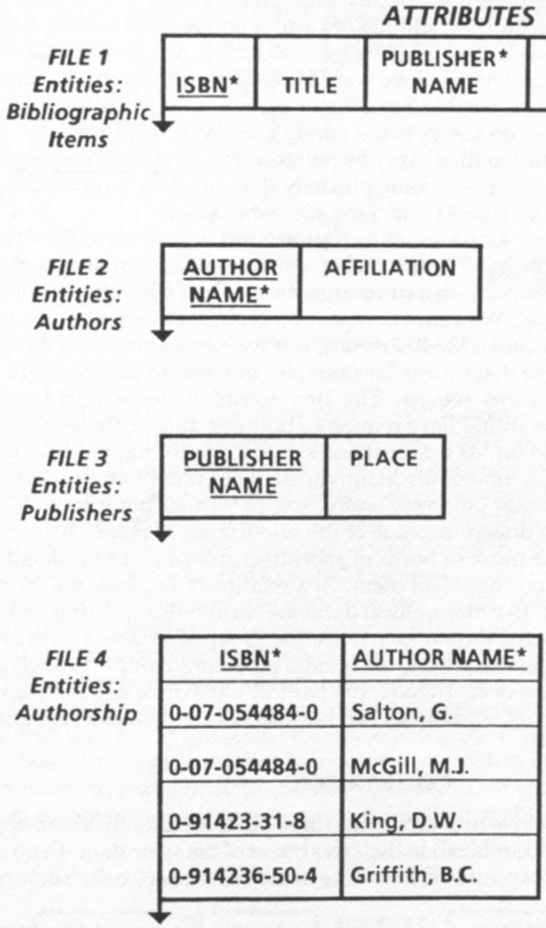


Fig. 27. Simple Bibliographic Data Structure Further Normalized to Accommodate Multiple Authorship (Repeating Groups).

handled in an implicitly hierarchical system by means of redundant hierarchies. The bibliographic record can be thought of as a hierarchical structure from one item to many descriptors and many authors, while the inverted index is a complementary hierarchy from author to many books, descriptor to many books, etc., as set out in figure 28.

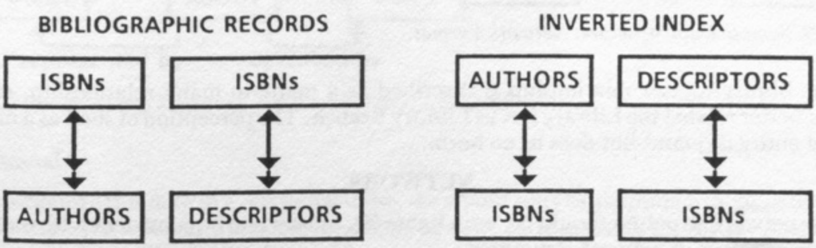


Fig. 28. Conventional Bibliographic Data Structures—Complementary Hierarchies.

The concept of repeating groups is essentially a hierarchical one, in which the children of a parent are thought of as a group of siblings of various number but similar description. When we try to put various levels of a hierarchy into one record, we are forced to adopt the expedient of repeating groups. That is, if we want to put descriptors into the same record as one that represents one bibliographic item, then we must make provision in some sense for repeating descriptors. If we do not try to put parent and children into one record, the so called repeating groups can be handled easily by pointers. In fact, the chains of pointers in a network structure (see figure 22) are doing precisely that: creating a repeating group of titles connected to Supplier A (Title #1 and Title #2), for example.

The idea of a MARC format, a data structure that in effect contains structure within the record, is an admirable one on many counts and of course an extraordinarily useful one. It has inherent limitations however. Such a structure is an excellent communications format, but it is not a database format. We can, via repeating groups, cram a one-to-many relationship (title to descriptors) into a MARC record, but we cannot accommodate a many-to-many relationship, titles to descriptors for example, in a record or we would have almost the entire data file in one record. The first record (title) would have several descriptors, those descriptors would have numerous postings, those titles would have still more descriptors, and on and on. As a consequence of this one-to-many limitation, a record structure such as MARC is inherently hierarchical. The concept of a repeating group then is a construct of a particular (and very useful) record format, but it is not fundamentally a construct of database design. Because of the importance of record formats such as MARC however, we tend to think in terms of repeating groups and to immediately ask how other data structures accommodate them. In a relational database we create a new entity to reflect the relations. In a hierarchical database we handle repeating groups either as we are used to them in MARC format (as a repeating group of adjacent records) or via a pointer structure. In a network database, we handle repeating groups through a pointer structure of connector occurrences. Indeed, the basis of a network structure is often described as the concept of the set, and the set can be described in turn as a repeating group linked by a pointer structure.

COMPARISON

It might perhaps be instructive to compare the three different data structure approaches (relational, network, and hierarchical) in their treatment of the same data. In an article on database systems, Atkinson presents the following example of a network structure[5] (figure 29).

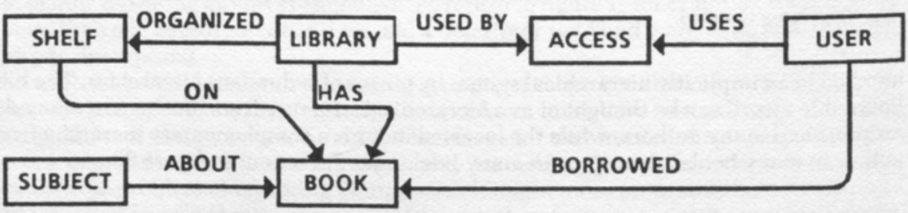


Fig. 29. Sample Data Structure, Network Format.

The library-to-user relationship is described as a many-to-many relationship, so we might better relabel the Library box as Library Branch. The perception of shelf as a meaningful entity is quaint but does us no harm.

NETWORK

The network structure should be, as in figure 30, redrawn to be consistent with our previous renderings of network structures.

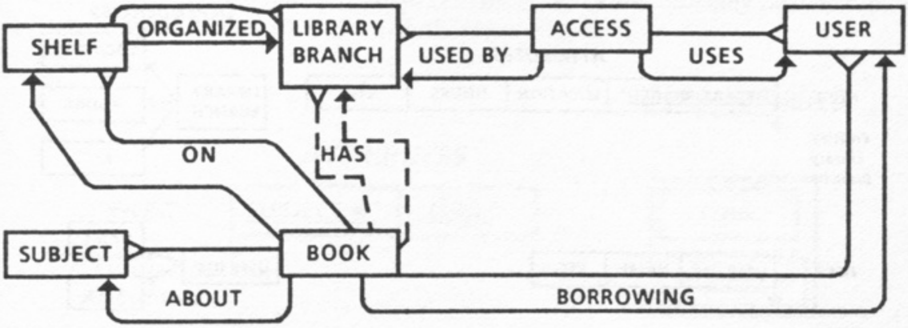


Fig. 30. Sample Data Structure, Network Format, Reworked.

In fact, the Has set is really not required, as the linkage between Library Branch and Book is provided by the Organized and On sets. Either alternative is really inadequate, however, as they allow one branch or one shelf to contain many books but not for one book to appear in more than one location. Surely we desire a system that allows for some duplication of holdings between branch libraries. To accomplish this we must insert a connector occurrence between book and shelf, which we might label Holdings. This way, one shelf can have many books, and one book can appear on many shelves (and thus in different branches). Now, however, the borrowed set must connect to holdings, as the user borrows a particular book from a particular branch library. Note, the addition of a Holdings connector occurrence record also allows for provision of multiple copies in the same library. The Holdings record could contain shelf number, ISBN, and copy number.

If we think of Subject as something akin to descriptor or subject heading rather than as a class number then the above model is again inadequate. (The model above allows the subject set About to contain many books, but not for a book to have multiple descriptors.) To handle a many-to-many book-to-subject relationship we would have to expand figure 30 to include an About connector occurrence. With these additions, figure 30 becomes figure 31.

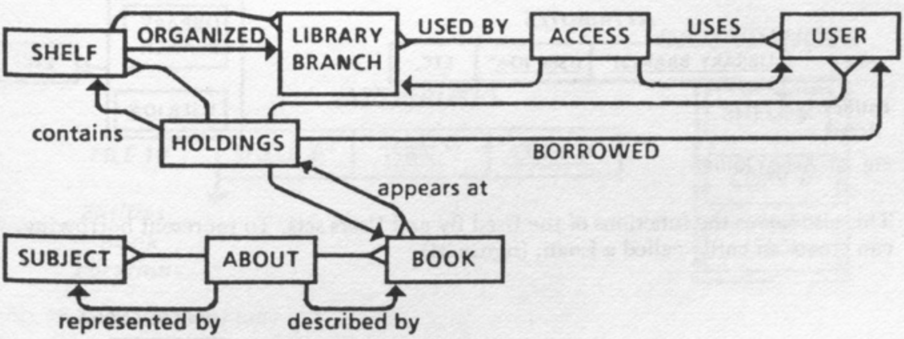


Fig. 31. Sample Data Structure, Network Format, with Additions.

Relational

To organize this data in a relational sense, we would start by defining a number of obvious entities. We will take the liberty of sketching those entities a little more fully in order to lend more verisimilitude. See figure 32.

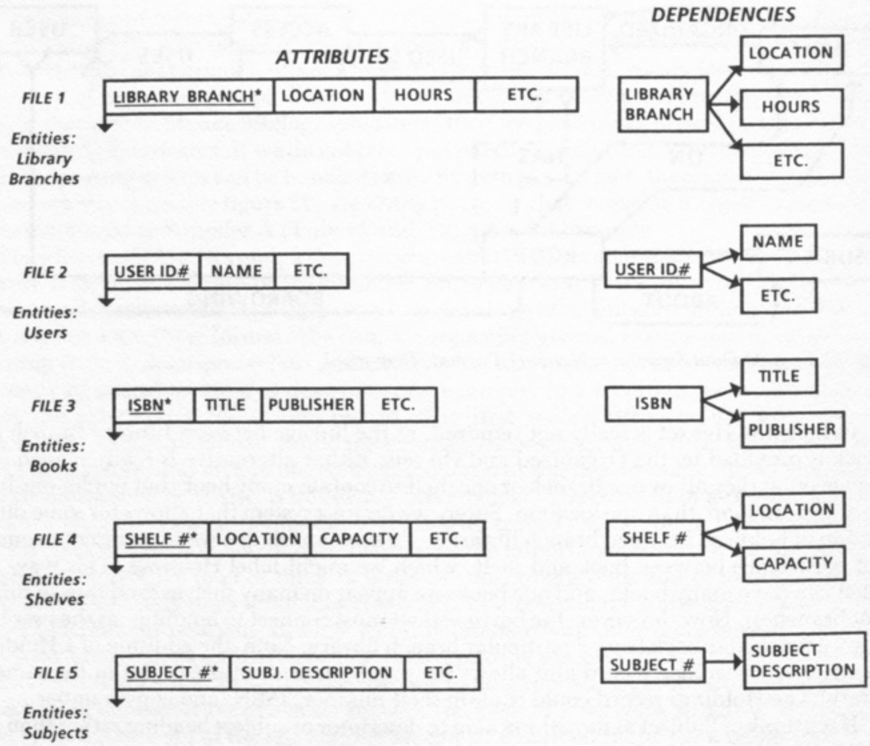


Fig. 32. Sample Data Structure, Relational Format, Basic Entities.

The intersection record Access becomes figure 33.

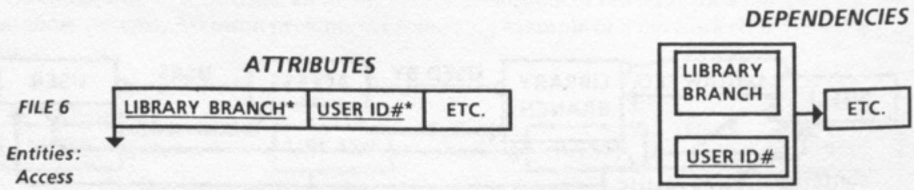


Fig. 33. Access Entity.

This also serves the functions of the Used By and Users sets. To represent borrowing, we can create an entity called a Loan, (figure 34).

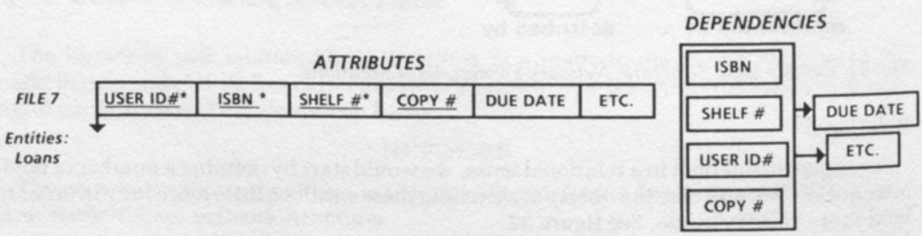


Fig. 34. Loan Entity.

To represent the relationship of subjects to books, as a many-to-many relationship, we can create an entity that we might call Aboutness (figure 35).

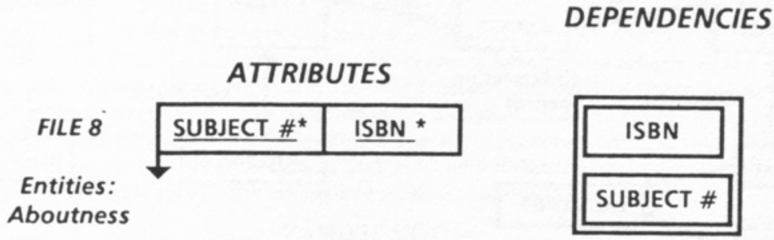


Fig. 35. Aboutness Entity.

To represent the Organized set we can create the entity Shelf Locations, and the Holdings connector occurrence record becomes the entity Book Locations, and this also serves the functions of the Contains and Appears At sets (figure 36).

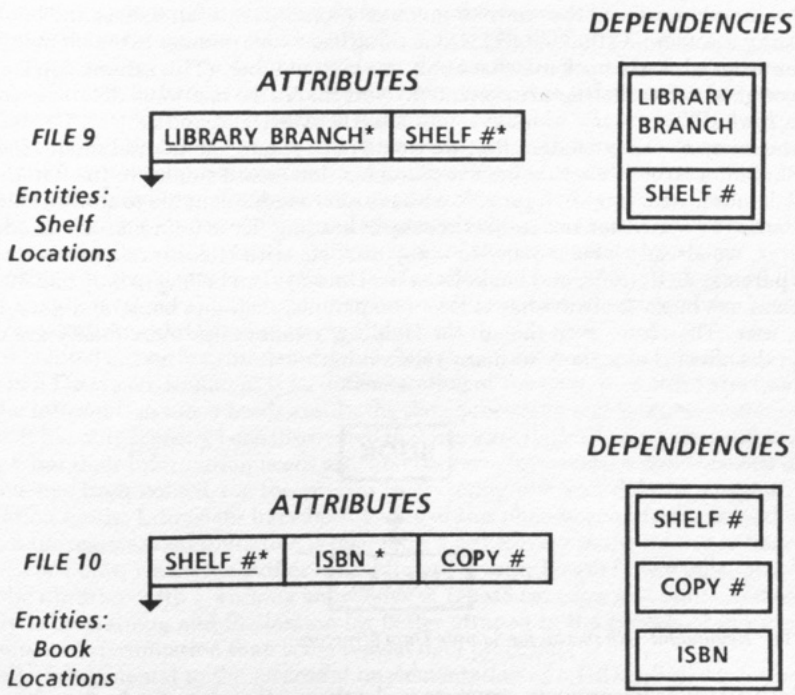


Fig. 36. Shelf Location Entity.

The Has set is redundant in a relational data structure. Files 9 and 10 above provide the links between library branches and books. The eight record types and eight sets of the network structure have become ten flat data files in the relational structure.

Hierarchical

Hierarchically, since this data structure has many-to-many relationships, it could be modeled in several variant forms (using redundant files, logical pointers, or intersection records). Perhaps the simplest conceptually is that in figure 37.

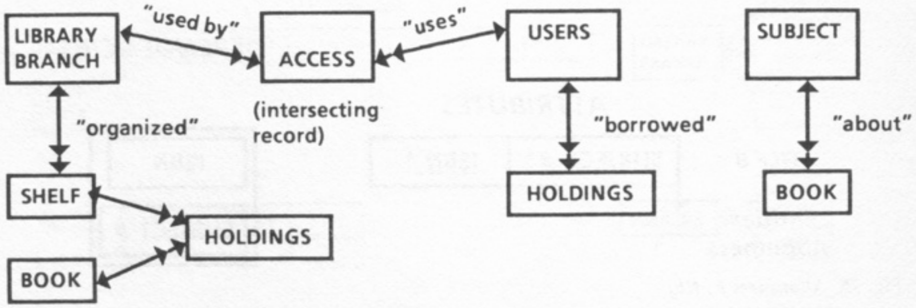


Fig. 37. Sample Data Structure, Basic Hierarchical Format.

We have again eliminated the equivalent of a Has set because of redundancy. We have had to create an intersecting record structure (Loan) to handle one many-to-many relationship (Library Branches to Users). Similarly, we have had to create the intersection record Holdings to represent the many-to-many relationship between shelves and books. We are also again assuming that Subject is something like a class number to which many books may be assigned, but a book may have only one class number. (This assumption is explicit in the original network structure presented.) Note that in the relational database structure above, file 8 allows a many-to-many relationship between book and subject. To accommodate the many-to-many relationship, we would have to add another redundant file, (or a logical pointer structure) or define a new intersection record similar to file 8 in the relational database structure. In figure 38, we have such a redundant file to allow a book to be represented by more than one subject or subject heading. Even without such an addition, however, we already have a many-to-many problem with Holdings since it is a child to three parents: shelf, users, and book. Even if a Holdings can belong to only one shelf, one user, and one book, it always has at least two parents, shelf and book, and potentially a third, user. Therefore, even though the Holdings relationships individually are one-to-many, the effect is of a many-to-many relationship.

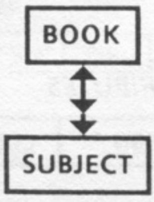


Fig. 38. "Redundant" Hierarchy for Sample Data Structure.

This particular library data structure is clearly one that does not lend itself well to a hierarchical representation, but it is by no means an unusual one. Indeed, almost any real-world library data structure must be substantially more complex.

Most complex data structures become too cumbersome to represent conveniently in a hierarchical data structure. Furthermore, many-to-many relationships almost inevitably occur; to cope with them additional hierarchies or logical pointers or intersection records must be added. Logical pointers and intersection records are far more flexible than additional hierarchies and easier to maintain, and, in turn, intersection records are more flexible than logical pointers as they can carry additional data as in figure 21. But given intersection records, one essentially has a network structure and one can regard the hierarchy as superfluous. Although a large number of DBMS built on a basically hierarchical structure, such as IMS, are still widely used, it is easy to see why hierarchical structures are almost universally eschewed in new systems development.

The real choice seems to be between network (often called CODASYL) structures and relational structures. The latter have the advantage of being more elegant in the sense of programming to manipulate them and have the further advantage at the implementation stage of tending to make the normalization process easier and more thorough because the constraints of the simple square table tend to force the decomposition of relationships to their simplest level. It is for good reason that most new database management system development is grounded upon the concept of a relational database.

RAMIFICATIONS

The most obvious ramification is the utility of DBMSs as the foundation for bibliographic information retrieval systems. Given greater knowledge of DBMSs, we can make better use of them in designing new systems and avoid needless duplication of effort. For this reason alone, it behooves us to know more about DBMSs.

Additionally, the world of online information retrieval systems is moving rapidly beyond the purely bibliographic into the realm of the numeric, from systems that directed us to document representations to systems that mount the data itself.[6,7] The bibliographic systems that we take for granted (OCLC, RLIN, DIALOG, SDC, BRS, etc.) are, to use our previous terminology, type 1 data structure systems that admirably direct us to documents, but in order to handle data and data relationships, type 2 data structure systems are needed, and we need to know how those systems are assembled. Using DBMSs we can not only more efficiently build systems that can manipulate bibliographic material than we can by building from scratch, we can build systems that are capable of far more than bibliographic data handling. As the cost of programming increases and the cost of storage decreases, the balance between from-scratch programming versus the use of externally acquired systems software constantly shifts in favor of the latter course of action.

In addition to the utility of DBMSs as the foundation of enhanced automated library and information retrieval functions, there are additional potent reasons why librarians should be familiar with DBMS concepts and terminology.

As we pointed out earlier, fundamental to the successful development of an information system in a DBMS environment is an understanding of the data: how it is to be used and what the information user's needs are. In the data-processing and business-systems community, it has only recently been discovered that users are typically not very good at articulating what their information needs are, and the term *information requirements determination* has been coined for the process of ferreting out and determining the users information needs. Librarians have been aware of this phenomenon for years and information requirements determination is really just a rediscovery of the reference interview process. The point, which the author has addressed at some length in two other articles,[8,9] is that the librarian with a working knowledge of DBMS concepts and online systems familiarity is by training and inclination far better attuned to the process of information requirements determination than is the typical data processor.

Equally fundamental to the successful implementation of a DBMS-based information system is the DD/D (data dictionary/directory). But the DD/D, particularly the dictionary component, is basically a thesaurus-like syndetic structure keeping track of the terminology and characteristics of the types of data stored in the system. It is quintessentially a library function. This point also is addressed at length in the same two articles.

Thus, not only are DBMSs important because library and information retrieval systems are increasingly being based upon them and most new bibliographic systems will be based upon them, they are important also because they contain classic library-like functions that can best be performed by persons with library skills and aptitudes. Those functions, however, have arisen in a data-processing environment, which has little awareness of the library/information technology environment and even less awareness of the contribution that persons from that environment can make. If those skills are to be utilized in the broader DBMS environment, then it is imperative that librarians/information technologists learn the concepts and the vocabulary of the DBMS environment. This article is a contribution toward that end.

REFERENCES

1. Richard W. Boss, "The ADABAS Data Base Management System," *Software Review* 1(2):175–77 (Oct. 1982).
2. William G. Town, John Powell, and Peter R. Huitson, "Recent Experience of the Application of a Commercial Data Base Management System (ADABAS) to a Scientific Data Bank (ECDIN)," *Information Processing & Management* 16(2):91–108 (1980).
3. Michael A. Huffenberger and Ronald L. Wigington, "Database Management Systems," *Annual Review of Information Science and Technology* 14:153–90 (1979).
4. C. J. Date, *An Introduction to Database Systems*, 3d ed. (Reading, Mass.: Addison-Wesley, 1981).
5. Malcolm P. Atkinson, "Progress in Documentation: Database Systems," *Journal of Documentation*, 35(1):49–91 (Mar. 1979).
6. Brian Aveney, "Electronic Publishing and The Information Transfer Process," *Special Libraries* 74(4):338–44 (Oct. 1983).
7. Michael E. D. Koenig, "The Information Controllability Explosion," *Library Journal* 107(19): 2052–54 (Nov. 1, 1982).
8. Michael E. D. Koenig and Stephen T. Kochoff, "The Education of Librarians for Data Administration," *Education for Information* 1(3):217–28 (Sept. 1983).
9. Michael E. D. Koenig, "Librarians; The Untapped Resource," *Datamation* 29(6):243–44 (Sept. 1983).

GLOSSARY OF ADDITIONAL TERMS FREQUENTLY USED IN DBMS LITERATURE

Cardinality: the number of rows of a flat table in a relation, that is the number of entities or tuples. (*See* ordinality)

CODASYL: an acronym derived from Conference on Data System Languages used to refer to the standards for a network data structure produced by that committee. Frequently used as synonymous with the term *network*, as in "hierarchical, CODASYL, and relational data structures."

Domain: the set of allowable values for an attribute. The string "califragalistic . . ." is not in the domain for the attribute ISBN.

Ordinality: the number of columns of a flat table in a relation, that is the number of attributes of an entity or tuple. (The Product File of figure 15, for example, has cardinality 4 and ordinality 5.)

Tuple: A term used meaning an entity and its attributes, equivalent to a row of a table in a relational table. A table is composed of tuples. ■■

News and Announcements

British Library Installs WLN Software

The British Library has installed Washington Library Network (WLN) computer system software at its computing service bureau, RHM Computing Ltd., located near London. The installation is the library's first step in major new systems development.

WLN's advanced online catalog database management facilities will assist the library in its goals of improving both the automated support for its internal operations and the information retrieval and cataloging services it provides other libraries. Several current MARC files will be combined into one large database, allowing database-wide (or "global") authority change and improved quality assurance.

The British Library will benefit from WLN's participation, as part of its shared development program, in the Linked Systems Project (LSP) sponsored by the Council on Library Resources. LSP will enable the library to communicate effectively with large American bibliographic systems such as those of the Library of Congress and the Research Libraries Group.

Now the national library of Great Britain, the British Library was established in 1974 by the merging of the British Museum Library, the British Lending Library, and the British Printed Bibliography and is one of the preeminent libraries in the world today.

WLN, a division of the Washington State Library, has been operating an online computer library system since 1977 and now services over two hundred participants principally in northwestern North America. The network began licensing library computer system software in 1979 and has several prestigious national and academic libraries participating as software licensees. ■■

RLG Announces IBM PC Version of RLIN Terminal

RLG is ready to distribute software that allows an IBM personal computer (PC) to operate like an RLIN-programmed terminal. This terminal emulation software is the first of three stages in RLG'S development of further extended character-set capabilities for RLIN.

The RLIN terminal emulation software, available on diskette, will enable libraries to use an IBM PC as a standard RLIN input terminal given the following IBM PC configuration:

- IBM PC system unit with 256 kilobytes (KB) of memory and two double-sided, double-density 360-KB disk drives (or an IBM PC/XT system unit with 256 KB of memory);
- standard IBM PC keyboard;
- IBM color/graphics adapter;
- composite video monochrome monitor (or IBM color monitor);
- IBM asynchronous communications adapter and cable;
- IBM synchronous data link control (SDLC) adapter and cable;
- PC disk operating system (DOS) diskette version 2.0, or later version

The software will support input and display of the full Roman alphabet, including the American Library Association extensions to the American Standard Code for Information Interchange (ASCII). These extensions include the double dagger, musical sharp, British pound sign, etc.

The second stage of development will be completed later this year, when RLG will offer support on the IBM PC for Cyrillic input, searching, and display in RLIN. Support for the Cyrillic alphabet will be based on the stage one RLIN terminal emulation software and the same hardware-software configuration.

Following this, in stage three, RLG will release software to support Hebrew (as well as Cyrillic and Roman) input and display on the IBM PC/AT. In 1986, support for Arabic will be added to the PC/AT software.

RLG already supports the input and display of Chinese, Japanese, and Korean vernacular in RLIN using specially designed "RLG CJK" terminals. ■■

Minds Meet On "Maggie III": Eyring Research Institute, Inc. Wins Library Contract

After seven months of researching systems and investigating proposals to build a customized computer system, Pikes Peak Library District (PPLD), has signed a contract with Eyring Research Institute, Inc. to build a new generation of "Maggie's Place," a Colorado Springs–developed library resource management system widely acclaimed for its computerized community resource files and home-user dial-up service.

The $1.3 million contract with Eyring calls for computer installation, customization of software, training, documentation, maintenance, and four 4 megabyte processors that have the power to grow 102 times more powerful than the current Maggie.

The new Maggie will use the Colorado Alliance of Research Libraries (CARL) online catalog and circulation system, making PPLD part of a statewide network that includes the Denver Public Library and libraries at the University of Denver, University of Colorado–Boulder, Auraria Campus, Colorado School of Mines, and the University of Northern Colorado at Greeley.

The CARL software and Eyring's conversion and upgrade of Maggie will run on Tandem's "Non-stop" processors. The Tandem machines total 16 megabytes, support 300 terminals and increase Maggie's disc storage three times.

Maggie's name affectionately comes from Margaret O'Rourke, a librarian who retired from PPLD in 1976, shortly after Maggie (the computer) was installed in the area that Mrs. O'Rourke's desk used to occupy.

Currently Maggie's Place handles book charge out and check in, purchasing and cataloging, budgets, payroll, decision support, word processing, electronic message, and mail. Over 1,700 people dial up to browse the files and hundreds access data daily from terminals in the library and branches. ■■

CAI Program Teaches OCLC Searching

The University of Florida Foundation has announced the immediate availability of *How to Search OCLC*, a computer-assisted instruction program. The program runs on an IBM PC. It has a large file of practice examples selected and matched at random, so the program is always different. Over 55,000 different author/title combinations are possible.

The program was designed by members of the Catalog Department at the University of Florida Libraries in an effort to cut costs and improve the quality and depth of training. It is aimed primarily at training students and clerical staff working in Technical Services, but has potential applications as a training course for new paraprofessionals and as a refresher course for more-experienced professionals and paraprofessionals in all areas of the library. It teaches many details not commonly encountered but important for accurate searching.

How to Search OCLC differs substantially from the CBT issued by OCLC with the M300 terminal. It runs on the IBM PC as well as the M300; the CBT runs only on the M300. Practice examples are always presented in a unique order. It features attractive, colorful graphics and optional sound effects. Users who do well on the game's comphensive "Grand Slam Exam" section are entered on the disc's "Hall of Fame."

System requirements are an IBM PC (at least 128K) with a single or double disc drive (double sided, double density). A color monitor is optional.

How to Search OCLC software and documentation cost $245 and are available from the University of Florida Foundation, Room 217 Library West, The University of Florida, Gainesville, FL 32611; (904) 392-0342. Send a stamped, self-addressed envelope for brochure. ■■

UTLAS Acquires Data Phase ALIS III

An agreement has been signed giving UTLAS Inc. exclusive rights to Data Phase's ALIS III software.

Data Phase Corporation is an automated library system vendor headquartered in Shawnee Mission, Kansas. ALIS III is the most advanced of a series of automated library systems developed by the company. ALIS III is a circulation system and online catalog based on the Tandem computer, which makes it compatible with the online CATSS system. ■■

Geac Develops Optical Disk Link for Micros

Geac has developed a product that allows personal computers to store and retrieve data from optical disks. Capable of storing up to 2 billion bytes of data on a single disk the size of a 33 1/3 rpm record, the Gigadisc holds as much information as a stack of floppies twenty feet high.

Geac's new product, called Gig-Attach, consists of a single circuit board and the software necessary to allow a PC to both read and write on Gigadisc under MS-DOS. Only one of the inexpensive attachments is needed to allow an entire network access to Gigadisc storage.

This development comes in the wake of Geac's agreement last November with Alcatel Thomson Gigadisc of France for the exclusive sale and distribution of Thomson Gigadisc optical disk storage systems in Canada. ■■

Blackwell Unveils Version 4

Blackwell Library Systems, Inc., marketer of the PERLINE™ and BOOKLINE™ systems, has announced the arrival of Version 4.

In addition to simplifying operations by centralizing all BOOKLINE™ and PERLINE™ capabilities in a single package with a single menu, Version 4 also offers multisite capabilities, publication flexibility, additional user shortcuts, and loan-control capabilities. The multisite capabilities allow each library to control its own ordering, receiving, and processing independently of other sites. PERLINE™/BOOKLINE™ Version 4 gives users the freedom to decide whether a given item is a serial or a monograph and then alter its designation upon order/receipt or vary it from site to site.

For more information, write or call Blackwell Library Systems, Inc., 202 East Main St., Suite 105, Huntington, NY 11743; (516) 351-1611. ■■

Subset of OCLC Database to be Available on BRS for Subject Searching

BRS Information Technologies and OCLC have signed an agreement whereby a subset of the OCLC Online Union Catalog will be available on BRS this fall for subject searching through the OCLC EASI (Electronic Access to Subject Information) Reference Service.

OCLC EASI Reference will enable BRS subscribers to conduct full-text searchers of approximately one million bibliographic records (without holdings information) from the OCLC Online Union Catalog. The OCLC EASI Reference is a dynamic database with an extensive online file of books, serials, sound recordings, scores, audiovisual materials, maps manuscripts, and software. It will have imprint dates within the last four years and will be updated regularly.

OCLC EASI Reference bibliographic information will be formatted into a public access/reference display familiar to BRS subscribers. Access to these modified records will be provided through numeric and title searchers as well as keyword and controlled vocabulary (Library of Congress Subject Headings and Medical Subject Headings) searches. The full-text searching capabilities of BRS, including full Boolean searching, will be available to OCLC EASI Reference users.

The BRS retrieval system enables users to store search strategies and to call them up for reprocessing at any time.

A dial-access terminal is required for access to BRS. OCLC members will be able to gain access to the OCLC EASI Reference Service from their M300 Workstations in dial-access mode.

All BRS subscribers, including both OCLC member and nonmember libraries, can use OCLC EASI Reference. ■■

OCLC To Offer Micro-Based Acquisitions System

OCLC is developing a microcomputer-based acquisitions system, ACQ350, which is scheduled for release in early 1986.

ACQ350 is being designed for use with the OCLC M300 Workstation, a modified IBM Personal Computer introduced in 1984. ACQ350 software will combine the advantages of microcomputing with access to the OCLC Online System.

Acquisitions functions best done at the local level, including fund accounting, file maintenance, and offline reports, will be part of ACQ350 software.

ACQ350 will maintain links to the Online System for bibliographic verification and name-address information. Users will be able to download bibliographic and address information and upload Direct Transmission (DX) information for electronic ordering.

ACQ350 will interact with SC350, the OCLC micro-based serials control system, and with LS/2000, OCLC's local automated library system.

Other planned ACQ350 features include automatic claims and cancellations, an allowance for hierarchical funds, use of default records for permanent constant data, accommodation of all MARC formats, and multiuser/multitasking capabilities. ACQ350 is expected to accommodate both small and large-volume acquisition users.

OCLC will support the online centralized Acquisitions Subsystem throughout the ACQ350 development and transition stages. OCLC and network staff plan to provide guidance to current Subsystem users who convert to ACQ350.

ACQ350 pricing and financial arrangements, exact date of availability, and the ultimate future of the OCLC Acquisitions Subsystem have not yet been determined.

MetaMicro Library Systems of San Antonio, Texas, is developing the ACQ350 software. MetaMicro Library Systems also developed the software for SC350, OCLC's microcomputer-based serials control system. ■■

New Circulation Module Online at Northwestern

On March 25, 1985, the main library at Northwestern University went online with the new NOTIS circulation module. The new application, like the rest of the NOTIS package, was designed and built by the automation staff at Northwestern University Library under the direction of Professor James Aagaard and his staff.

The new module offers completely online file updating for all of the circulation functions. It supports both regular and hourly checkouts, using machine-readable identifiers for patron and material identification. The software is designed to support the complete physical inventory of the library, including bound and loose issues of journals.

The module is integrated with the rest of the NOTIS application, including acquisitions, serials control, authorities, database maintenance, and the online catalog.

Northwestern is using laser scanners for the circulation of materials, and is printing due-date slips at the charge desks. The Northwestern file has approximately 2 million items. ■■

CORRECTION, JUNE 1985

The name of the author of Reports and Working Papers, "1984 Automated Authority Control Opinion Poll: A Preliminary Analysis" was incorrectly listed on the cover and contents page of the June 1985 issue. The author's name is Barbara B. Tillett.

Recent Publications

Special Review

In an effort to create better links between automation specialists working on both sides of the Atlantic, the editors of Information Technology and Libraries *and* Program *agreed to exchange issues for review in each other's journals. To look at and describe* Program *for American readers,* ITAL *commissioned Michael Gorman, who has held influential library positions in Great Britain and the United States.*

Program, the London-based society Aslib's journal of automated library and information systems, has been published quarterly since 1966. In that time, the journal has undergone three major changes of format (in 1968, 1970, and with the first issue of 1985). This rate of change seems excessive even for a journal in the field of librarianship. The title has remained the same (though the subtitle "automated library and information systems" was intermittently present in some earlier editions), so this journal does not qualify for the "Snake-in-the-grass" award bestowed annually by RTSD/SS. However, such changes are anathema to all right-thinking folk, especially when breezily announced, with neither explanation nor excuse, as was the latest change in *Program* in vol. 19, no.1. For the purpose of this review I have studied all four issues (v.18, nos. 1–4) of 1984 and the two issues (v.19, nos. 1–2) of 1985 available at the time of writing.

Program is, at this time, edited by Lucy Tedd, who worked at one time at the College of Librarianship, Wales, and is now a free-lance writer, editor, and consultant. She has one of those delicious Welsh addresses that looks like a series of anagrams from a Czech crossword puzzle and sounds, if pronounced correctly, like a poem. It is, for those who wish to contribute to *Program*, Llys Blodau, Iorweth Avenue, Aberystwyth SY23 1EW, Wales.

The journal has a ten-person editorial board consisting of academic, college, special, and public librarians (all British); rumly enough, they are all men.

In its present manifestation, *Program* is printed on good quality paper, perfect bound in soft covers, and, over the six issues under consideration, averages approximately 100 pages an issue. It costs £30 annually in the U.K. and $69 in the U.S. The contents of *Program* fall into five categories—articles, short communications, news items, book reviews, and short notices of "other items received." The six issues under review contain 27 articles (an average of 4.5 an issue), 24 short communications (an average of 4 an issue), 68 news items (an average of 11.3 an issue), 43 reviews (an average of 7.16 an issue), and 12 short notices of items received (an average of 2 an issue).

The 27 articles published in this period average 12.9 pages each, with the longest article being 30 pages long, and the shortest being 9 pages long. They are overwhelmingly British in origin. All but 6 are by British authors (the others emanating from Australia (2), the United States (2), Denmark, and the Netherlands). This might not surprise anyone were it not for the fact that half of the circulation of *Program* is outside the United Kingdom. The articles are heavily concerned with particular applications of automation in libraries. Again, the British emphasis can clearly be seen. Fourteen of the 27 articles are concerned with specific applications in British libraries. Half of the articles of foreign origin listed earlier in this paragraph are concerned with particular applications in their countries of origin. The exceptions are a survey of library automation in Scandinavia and surveys of microcomputers in Australian and U.S. libraries (the latter is by

Ching-chih Chen). The article on a particular U.S. library application concerns the ubiquitous Pikes Peak system.

In addition to the concentration upon particular systems, two other areas of concentration can readily be seen. Eleven of the 27 articles deal with microcomputer applications, general and specific, and 4 deal with database management systems. (Naturally, these numbers overlap, since, for example, an article concerned with a microcomputer application in a specific library shows up in both sets of figures.) The particular libraries and library systems described in these articles show an impressive and interesting range. They include a national library (the British Library Lending Division), a cathedral library (that at Canterbury), special libraries (Harwell Atomic Energy Research Centre, ICI Organics Division, British Aerospace), a medical library (that of the Radcliffe Hospital in Oxford), academic and college libraries (University of Aston, University of Tasmania, Leicester Polytechnic), and public libraries (those of Bedfordshire and Rotterdam). This range compensates for the Britocentrism to some extent, since it guarantees relevant comparisons to the U.S. reader even though the articles do not concern U.S. applications. The descriptions are, in general, informative and free of self-congratulation and jargon (the besetting sins of the "how-we-did-it-good" article).

The articles that are not concerned with specific applications deal with a range of topics. Apart from the microcomputer and DBMS articles mentioned earlier, topics dealt with include an outline of a plan for an expert system for AACR2 cataloging and searching in a free-text system. The microcomputer articles include national surveys, a description of a microcomputer database management system, and current awareness microcomputer systems.

Quite arbitrarily, I have set a minimum of five references as the criterion for the assessment of the articles as being research-oriented. Eleven (41 percent) of the 27 articles meet this standard. On an even more subjective level, I would judge these articles to be no better and no worse written than their American counterparts. I realize that this is praising with faint damns, but also realize that not even the most scintillating literary stylist could do much with, say, the topic of database management systems. To embark on an odious comparison, I would say that the articles in *Program* are less research-oriented than those in *ITAL*, are longer, are concerned more with particular applications than with generalities, and are more local in that they deal with specifically British topics to a greater extent than the *ITAL* articles deal with specifically American topics. That being said, I would recommend these articles to any serious reader of *ITAL* because the range of experience that they depict has lessons for American libraries and because, despite superficial differences, both journals are about the same business.

The short communications are, naturally enough, concerned almost exclusively with the reporting of practical experience in the field. Topics, chosen at random from the issues under review, include "The BLEND Network and Electronic Journal Project," "Eighth Intermarc Software Subgroup [now European Library Automation Group] Meeting, a Report," "Retrieval with dBase II," and "A Computerised Union Catalogue of Literary Manuscripts." They occupy an average of slightly under 20 pages in each issue, and range from 2 to 6 pages in length. Here again we see an impressive range of topics, many of which are of great interest to the American reader, despite the fact that most are concerned with British or British-centered matters. Four of the 24 short communications deal with events and matters outside the U.K. They concern libraries or meetings in Nigeria, France, Italy, and the United States.

The 68 news items occupy, neatly enough, 68 pages. I will insult neither *Program* nor *ITAL* readers by giving the average number of pages devoted to news items. In any event, they are the usual mixture of old news, good news, non-news, and commercial puffery. We learn that OCLC has acquired Avatar Systems, that the EUDISIC Annual Conference has been announced, that SCICON operates an energy trade databank, that Dr. Phil Holmes has been appointed Director for Information Services for something called Jordans, that *Books in English* is to be cumulated for

1971–80, that there have been changes in the IFLA UBC office, and that there will be an information network established at the University of Papua New Guinea. I leave the gentle reader to decide which of these news items belongs in which of the categories set forth earlier in this paragraph. Personally, I think all of them are a frightful bore, and, in the unlikely event that I were made the editor of one such learned journal, would either curtail severely or eliminate this section. I hasten to say that the preceding is a general stricture, and that *Program* is no worse in this regard than any other journal.

The book reviews in the issues under consideration present rather a different picture from that of the areas previously described. The 49 books and serials reviewed in the 43 reviews are all in English, but not even the majority emanate from the United Kingdom. The country of origin of the books reviewed is, as far as can be determined, United States (25), United Kingdom (19), West Germany (2), and Canada (1), Israel (1), and Australia (1). The nation of my birth and citizenship and the nation of my residence may be divided by a common language, but the shadow cast by American publishing in the field of library automation is great indeed. The reviewers are all British, but they exhibit no discernable bias against the monstrous regiment of U.S. books. The reviews are, on the average, slightly more than one and one-half pages long, and are generally descriptive rather than critical, open-minded rather than opinionated, and have an eye on the practicalities of cost-benefit.

The short notices of "other items received" are OK, I suppose, but I really do not see the point. Review them, or list them, or leave them alone is our simple, rugged midwestern creed, and library journals would be more wholesome if it became the general practice.

Program is a good buy and a good read, has much to commend it, and should be familiar to any serious American student of library automation. With *ITAL* and *Program* on the reading list, there is little that would escape one's attention.—*Michael Gorman, University of Illinois at Urbana-Champaign.* ■■

Reviews

Hedstrom, Margaret L. *Archives & Manuscripts: Machine-Readable Records.* SAA Basic Manual Series. Chicago: Society of American Archivists, 1984. 75p. ISBN: 0-931828-60-0, softcover, $6 to SAA members and $8 to others.

The logistics of handling and preserving machine-readable records present a series of new and difficult challenges for the archival profession. Computer-generated data is an ever-increasing component of the documentation created by contemporary organizations, offices, and individuals. The wide variety of automated records and systems, coupled with the ever-changing nature of information technology, places an unusual burden upon archivists, particularly those lacking any day-to-day experience in this arena. As archivists begin to apply existing and newfound techniques, skills, and expertise to these materials, they must be familiar with certain basic principles and characteristics that apply to machine-readable records.

With this in mind, the newest publication in the Society of American Archivists Basic Manual Series attempts to identify, define, and detail the ramifications of machine-readable records for the archival profession. Margaret L. Hedstrom has created a useful and practical reference tool for archivists and other professionals wishing to grasp the fundamentals of automated records. The manual outlines procedures to "locate, appraise, accession, process, and preserve machine-readable records." More complex and contemporary facets of information technology are not addressed, however, with the caveat that "archivists have not developed techniques for preserving some of the most advanced and technologically sophisticated machine-readable records," and that "archivists must progress from handling relatively simple records to tackling a wide range of more complex problems."

The manual contains a preface, introduction, four chapters, a glossary, and a select bibliography. The latter two components are useful inclusions for the intended audience. The document is thoughtfully arranged, with several well-placed examples, charts, and illustrations. Hedstrom also controls her use of computer terminology for the benefit of the uninitiated reader. The presence of examples depicting actual machine-readable data also succeeds surprisingly well.

The four main chapters of the manual present an introduction to the world of computers and information/data processing, a discussion of the arrangement and storage of machine-readable records, an overview of management and preservation practices, and a thoughtful essay on the office of the future.

Hedstrom's charge to the archival profession, to begin to manage automated records now or face the "potential obliteration of significant portions of the historical record," is a painful reality for many archivists. The author believes that the multitude of problems posed by machine-readable records, including the "proliferation of information in a variety of formats and on different media," should be countered by applying the modified but traditional archival practices of appraisal, arrangement and description, records management, and deaccessioning. Hedstrom also stresses the crucial need for archivists to inject themselves into the life cycle of automated records at the earliest possible moment.

Basic background information necessary to handle machine-readable records is presented in chapters 1 and 2. Chapter 3 contains a wealth of information on management and preservation strategies. All archivists will benefit from a thorough understanding of this material. The discussion of records surveys, appraisal guidelines, preservation procedures, adequate documentation, and appropriate access to these materials provides archivists with a framework for action.

On the topic of preservation, Hedstrom states that "no magnetic storage medium meets, or even approaches archival standards for permanent preservation of data." The current alternative—creating a security back-up file for automated records—is not noted for some twelve pages. Such observations, however, do not detract from the overall utility of this document.

As an introduction to the topic of machine-readable records, and as a ready reference source on the subject, this manual fills a long-standing need and should prove an invaluable aid to archivists. For the archivist faced with his or her first group of machine-readable records, or for the archivist contemplating an initial and/or expanded role in handling automated records, this manual provides sound analysis and recommendations. The profession as a whole will gain from the creation of a solid reference source on which to build future work.—*William E. Brown, Jr., Yale University, New Haven, Connecticut.* ■■

Other Recent Receipts

Listed here are books and other publications of potential interest to members of LITA, received for review. Some of these materials may be reviewed in later issues of ITAL.

Clark, Philip M. *Microcomputer Spreadsheet Models for Libraries: Preparing Documents, Budgets, and Statistical Reports.* Chicago: American Library Assn., 1985. 118p. ISBN: 0-8389-0403-3, softcover, $24.95.

Computer-Readable Databases: A Directory and Data Sourcebook. [V.1] *Science, Technology, Medicine,* [V.2] *Business, Law, Humanities, Social Sciences.* 1985 ed. Chicago: American Library Assn., 1985. 1658p. ISBN: 0-8389-0416-5 (V.1), 0-8389-0417-3 (V.2), 0-8389-0415-7 (2 vol. set), softcover, $157.50 (set) or $87.50 each. Issued previously by the American Society for Information Science (ASIS) and Knowledge Industry Publications. Martha E. Williams, editor in chief.

Curley, Arthur, and Broderick, Dorothy. *Building Library Collections.* 6th ed. Metuchen, N.J., and London: Scarecrow, 1985. 339p. ISBN: 0-8108-1776-4, hardcover, $18.75.

Directory of United Nations Databases and Information Systems 1985. New York: United Nations, 1984. 323p. ISBN: 92-9048-295-8, softcover, $35. UN Sales No. GV.E.84.0.5. Compiled by the Advisory Committee for the Coordination of Information Systems (ACCIS).

Gaber, Walter A. *PC Abstracts: Abstracts and Index of Periodical Literature for the IBM PC and PC Compatible User 1984.* Garland Reference Library of the Humanities, V.599. New York and London: Garland, 1985. 529p. ISBN: 0-8240-8720-8, hardcover, $60.

Katz, Bill, and Fraley, Ruth A., eds. *Evalua-*

tion of Reference Services. New York: Haworth, 1984. 334p. ISBN: 0-86656-377-6, hardcover, $29.95. Also published as *The Reference Librarian*, no. 11, Fall/Winter 1984.

Matthews, Joseph R. *Directory of Automated Library Systems.* Library Automation Planning Guides Series, no.2. New York and London: Neal-Schuman, 1985. 217p. ISBN: 0-918212-82-0, softcover, $34.95.

Mount, Ellis, ed. *Data Manipulation in Sci-Tech Libraries.* New York: Haworth, 1985. 131p. ISBN: 0-86656-441-1, hardcover, $19.95. Also published as the journal *Science & Technology Libraries*, V.5, no.4, Summer 1985.

Saffady, William. *Video-Based Information Systems: A Guide for Educational, Business, Library, and Home Use.* Chicago: American Library Assn., 1985. 240p. ISBN: 0-8389-0425-4, softcover, $30.

Scientific and Technological Information for Development. Proceedings of the Ad-hoc Panel of Experts on Information Systems for Science and Technology for Development held in Rome, Italy, 21–25 January 1985. New York: United Nations, 1985. 179p. Softcover. United Nations publication Sales No. E.85.II.A.7.

Spreitzer, Francis, ed. *Microforms in Libraries: A Manual for Evaluation and Management.* Chicago: American Library Assn., 1985. 63p. ISBN: 0-8389-3310-6, softcover, $8.95. ■■

Letters

To the Editor:

First, my congratulations on another fine issue, ITAL 4(1): 1985 March. Janet Asteroff's "On Technical Writing and Technical Reading" was a special pleasure; it should encourage all of us who write to do our best to retain a "human voice" rather than lapsing into the safe, bland "professional" voice.

I have some updates to "The RLIN command Analysis System." We now use the system to test and validate CPU usage for typical sets of commands. (This use came into play in January 1985 when RLG moved from an IBM 3081 to an Amdahl 5860.) Figure 11, marked as "modified", represents the current form of that particular chart, which now starts at 60 percent rather than 50 percent. We are now logging 5 percent of sessions rather than 10 percent, using a different algorithm which is equally pseudorandom in its effects. Given increased use of RLIN, 5 percent logging still gives us 10,000–15,000 logged commands each day.

Finally, some readers may have wondered why we don't log the *result* of search commands, that is, the number of records found. We're now considering a change to provide that information. Certainly, an online catalog logging system should include number of records found as part of logging, so that search strategies can be studied and evaluated. RLIN isn't an online catalog, and the initial purposes of command logging didn't include studying search strategies, but we're getting more out of the command analysis system as we use it.—*Walt Crawford, Manager, Product Batch Group, The Research Libraries Group, Inc., Stanford, CA.* ■■