Selection of Software for Patron Use in Libraries

PEGGY SEIDEN

ABSTRACT

THE LIBRARIAN WHO SEEKS to develop a collection of computer files seems faced with a formidable and often unfamiliar task. This article provides an in-depth study of the selection process for patron-use software programs and data files. It begins with an analysis of the unique characteristics of software—its function, its format, and its mode of publication/distribution. The author discusses whether selection and acquisition can be integrated into existing practices or must be treated in a unique manner. The article covers aspects of the selection process, including the development of a collection policy, the assignment of selection decisions, methods of financing purchases, selection criteria, and the evaluation of individual pieces of software. Sources of information—descriptive, evaluative, and anecdotal—on computer files are listed and described. Acquisition and collection evaluation issues are also discussed.

INTRODUCTION

The librarian who seeks to develop a collection of computer files is faced with a formidable and often unfamiliar task. There are probably over 100,000 microcomputer programs "in print" and a large and ill-defined set of finding and evaluation tools which cannot hope to keep up with the volume. In addition to this unfamiliar body of information with which the software selectors must acquaint themselves, there are technical issues such as system compatibility and legal issues such as licensing and copyright restrictions.

Peggy Seiden, Library, Pennsylvania State University, 3550 Seventh Street Road, New Kensington, PA 15068
LIBRARY TRENDS, Vol. 40, No. 1, Summer 1991, pp. 6-41

Furthermore, the cost of the software is often ten times or more what one would spend on a book, so the pressure to make correct collection decisions is much greater than when dealing with print materials. But in the time it takes the conservative selector to do a thorough evaluation of one software program, a new version may be published. Nor can one rely on printed reviews since there are many types of software for which little information exists. Even the software purchasing process is difficult to integrate into existing acquisition procedures, since the software industry still seems relatively unaware of libraries as a potential market for the products.

Nevertheless, though much about software selection is new and different, there are many aspects that require the same treatment as librarians give to other types of materials. This dichotomy between what is familiar and what is not has led to two schools of thought among librarians about how to treat these materials (Johnson, 1990, p. 7). While one school of thought says that computer files require a totally new approach, new staffing, new policies and procedures, the other thinks that computer files should be integrated into the existing library structure.

In truth, the choice is not so black and white as it first appears. The nature and types of computer files are very complex, and this complexity requires that the library use a mix of approaches in selecting these materials. This article presents an overview of the selection and acquisition process for the types of computer software which libraries typically collect for use by their patrons. Computer software is defined broadly to include not only software programs, but data files and other types of electronic resources as well.

Background

Most of the literature concerning the introduction of the microcomputer as a library public service was written in the early to mid-1980s. A review of the library literature since this time reveals few significant articles or books on selecting and acquiring software for public access. The literature that does exist falls into several categories. The largest category of material on software are reviews of individual software programs or types of programs. There have also been a considerable number of articles and chapters both in the library literature and the educational technology literature on evaluating software and selection criteria. These articles are referred to in the section on evaluation. Other publications on selection provide lists of resources to aid in selection, but many of these are now outdated. Some of the more recent ones are referred to in the section entitled "Sources of Software Information."

However, there is little written which provides a holistic treatment

of this subject. Sheila Intner (1988) provides an excellent overview of major issues and concerns from a management perspective. She raises many useful questions which must be asked by those planning to build a software collection.

Other valuable articles and books are cited throughout this article. However, many selection issues are peculiar to different types of libraries. The following sources are recommended for a more in-depth treatment of issues pertaining respectively to academic, school, and public libraries.

Katherine Chiang and Howard Curtis (1987) of Cornell's Mann Library have authored two chapters in *Public Access Microcomputers in Academic Libraries* which are case studies of the Mann Library, in which the authors discuss, in detail, the framework that the library established for selecting and acquiring computer files. Margaret Johnson's (1990) article in a recent RLG publication, *Computer Files and the Research Library,* provides a broader view of collection management and development. Though Johnson defines computer files to include both data and program files, much of the article focuses on the former. Johnson exhorts research libraries to take the lead in their institutions in computer file collection and management, but notes that if libraries are to assume this role, they must meet various challenges, the greatest and foremost of which is defining the universe of computer files. This is particularly difficult since many computer files are unpublished. Johnson emphasizes the need for coordination both within the library and within the university if computer files are to be successfully integrated into the research university.

Askey (1987), Swigger (1986) , and Clyde and Joyce (1985) discuss developing software collections in school library and media centers, where selection decisions are tied closely with curriculum decisions. While neither Askey nor Swigger distinguish between collecting for the library or the classroom, Clyde and Joyce address issues specific to the librarian or media specialist. Judith Askey's chapter on selecting software for school libraries focuses on selecting software to meet specific learning goals, but Swigger cautions against defining a collection by the curriculum only. If software has the potential to revolutionize the curriculum, the content of the library's collection must be broader than that of the curriculum (p. 284).

Patrick Dewey's (1984) publication, *Public Access Microcomputers: A Handbook for Librarians,* not only defines the selection process, but provides examples of appropriate software for public libraries as well as key resources to help librarians find software. Dewey (1990) has recently updated this publication.

Defining Computer Files

A useful first step in developing a collection development policy is to define the types or categories of software the library will or will not collect. This requires a basic understanding of the medium and the possible uses of computer files (Johnson, 1990, p. 4). As Chiang (1987) notes, there are a number of existing taxonomies of software (p. 38). Dewey (1988) lists four categories (business, education, games, and utilities) (p. 65); Intner (1988) defines twelve categories of software by subdividing business; these groupings can further be subdivided by expertise and subject (pp. 9-10). Chiang (1987) and her colleagues at Cornell developed a taxonomy based on library functions (pp. 38-47). While these approaches may be useful in some contexts to adequately encompass all types of computer files, it seems that a somewhat different tact is needed. Electronic resources can be defined or classified by three parameters—function, their format, and the mode of publication and/or distribution.

Functional Categories of Software

Broadly speaking, electronic resources can be divided up between software programs and data files. Within both these categories there are many different subcategories.

Software Programs. These are programs or procedures which operate the hardware and manipulate data and include operating systems, programming languages (assemblers, compilers, interpreters), utilities, applications and subroutine libraries. Generally, libraries have been and continue to collect the following types of applications (including subroutine libraries and programming tools):

Personal Productivity—This is certainly the best known type of software. Sometimes called business software, the usefulness of these applications goes well beyond their use in business contexts. Generally, these programs are used to enhance personal productivity. This category includes products like word processing programs, spreadsheets, database management systems, and communications software. Other types of software include graphics packages, outliners, presentation packages, equation solvers, page layout programs, or mailing label production packages.

Professional Software—These are programs that are specific to particular professions such as accounting programs, real estate programs, architectural CAD packages, and engineering programs.

Courseware—This software is developed specifically for use in instruction including drill and practice, simulation, and modeling programs. It includes traditional computer-assisted instruction

programs as well as innovations using hypermedia. Software used in grades K-12 is substantially different from that used in higher education. The K-12 market is more mature and the curriculum better defined. This category also includes training materials used in corporations and the military.

Research Applications—These programs may include data gathering software, statistical analysis tools, sophisticated numeric and mathematical programs, and modeling programs. This software is largely found in university or corporate environments.

Entertainment Software—High and low resolution graphics and textual programs including arcade type games as well as more educationally focused programs are typical of this genre.

Programming Tools and Languages—This type of software is used to create programs. The category includes compilers like C and Fortran, interpreter like LISP and BASIC, authoring languages, authoring systems, and programs like HyperCard (although this type of program also falls into the personal productivity category).

The lines between these functional categories are difficult to draw. For example, a tool used by one person for research or in their profession could be used by another for teaching.

Data Files. Software programs manipulate information or provide information—they *perform a function.* But data files contain information and hence are much more like other types of library materials. The information in data files can be structured or unstructured; it can be textual, numeric, graphic (still or animated), or sound. Some data files are sold with a software program which allows the user to access, manipulate, and retrieve the data. The categories listed below are based on traditional classes of library materials with which librarians have long been familiar.

Bibliographic Data Files—These may be databases on CD-ROM, disk, or tape. Libraries can develop or acquire bibliographies, indexes to library collections such as songs, newspapers, local documents, or other special collections in machine-readable format.

Reference Sources—This group of files is growing at a rapid rate and includes full text of encyclopedias, dictionaries, statistical compendia, etc.

Secondary Sources—Electronic journals are just beginning to emerge as a new type of publication. The electronic journal has evolved along several distinct paths. Some journals have evolved from edited online bulletin boards; others were conceived of as electronic versions of existing print journals; still others look and behave like print

journals but are available only in machine-readable form. Although many books are produced from digital copy, publishers are only starting to experiment with electronic distribution. Though one cannot curl up with a computer-based book in bed, the Japanese are moving ahead in this area and the United States will probably soon see similar products. Already a consortium of universities and Xerox is experimenting with local publication (using a laser printer) of materials stored elsewhere, and such technology holds much promise for libraries facing future space shortages.

Primary Sources—Collections of literary texts such as the Oxford English Text Archives and efforts to develop an archive for texts in the humanities in this country have received some recognition. But many literary texts have been translated into digital form by individuals and are not published as such.

Data collections, which may be numeric, like census data, or economic data, or textual like genetic or meteorological databases, are increasingly common in the social and hard sciences.

Format

Most of these types of program and data files are available in a multiplicity of formats. One tends to think of software as the "stuff" on floppy disks. It is widely recognized that punched cards and paper tape are defunct technologies, but the ever evolving computer storage technology has left even more recent innovations, such as the microcomputer cartridge and 8 inch disk, molding on library shelves, and the $5^1/_4$ inch disk will probably soon be obsolete as well. In addition to floppy disks, software and data files are distributed on various compact storage devices including CD-ROM and videodisc. Workstation class machines use a variety of storage devices including cartridge and tape, and mainframes continue to require magnetic tape as a medium for offline storage of data.

All of these technologies can be thought of as "stand alone" and, in selecting and acquiring these materials, the library or other computing facility physically houses these materials. However, electronic resources can also be delivered from sites remote to the library over data and phone lines through networks or via modem/telephone access. Libraries have long been familiar with access to the major bibliographic utilities or commercial database services through networks and modems. But there is a rapidly growing body of other types of information resources which can be accessed remotely. The national and international data/telecommunications infrastructure permits high speed access to gigabytes of data stored virtually anywhere on these networks. The technology to deliver information

electronically to personal computers at home from across the county or across the country is available today. Many research and educational institutions are taking advantage of this infrastructure to begin "publishing" locally created information resources over national networks. Some of these resources are informal in nature, like many online bulletin boards, but other resources are the equivalent of printed published materials. For example, Virginia Polytechnic Institute recently announced the electronic publication of an international journal on hospitality. As the digital library grows, it means that a collection of a library no longer is limited to the physical space of that library. While this type of resource is a boon to libraries who are constantly fighting for space, these resources raise questions about the responsibility of the library for materials outside the library's walls. In this case the selection of materials for use by library patrons may preclude physical acquisition but require that the library catalog these sources so that their patrons are aware of them or, in certain instances, purchase accounts so that patrons can access these resources (Chiang, 1987, pp. 41-42).

Mode of Publication/Distribution

The third parameter which characterizes electronic media is the mode of publication or distribution. A wide variety of paths are available to get software from the producer to the user. Since digital formats not only are easy and cheap to reproduce, but also the copy of a program or data file is as good as the original; software does not have to be formally published to be distributed or used. Commercial publication mechanisms for software are roughly similar to those for print—i.e., large publishing houses, smaller specialized publishers, and even alternative presses. Computer files may be published by vendors exclusively dealing in software, by hardware vendors, and by traditional print publishers. A great deal of software, though, is available from noncommercial sources. Some of these publishers—universities, associations, the government, and clearinghouses—use retail strategies, but some use public domain strategies (Seiden, 1988 pp. 2-3).

The computing/hacker culture from which the current software industry developed is based not on a "market economy," but on one in which the free and open exchange of information and software is considered necessary in order to move the "art" forward. Steve Erhmann (1987), technology program officer of Annenberg/CPB, has called this public domain distribution mechanism a "circle of gifts." Eventually the developers who place their software in the public domain receive some other program in return.

When a program is placed in the public domain, the author

relinquishes control over that program by not copyrighting it. Public domain programs are generally written to satisfy a need of the author who chooses to share the program with others rather than sell it. The software may be distributed via user groups, libraries, and electronic bulletin boards. Shareware or user-supported programs may also have been written for the personal use of the programmer, but the author may request donations from users, if the users, after trying the program, decide to keep it. Some authors are casual about the process, but others may support their programs full time. In the case of user-supported software, the users are expected to provide the authors with suggestions for improvements to the programs. Some programs are developed to the point where they are equal or better than similar commercial programs. To encourage payment, some authors ask users to register. The registration fee guarantees support, documentation, and updates (Seiden, 1987, pp. 10-11).

The price of these voluntary contributions has increased to the point where some shareware packages cost as much as their commercial counterparts. Other software has gone commercial, and some software collections are being packaged and sold for profit by third parties. These individuals and organizations add value to the collection in some way. They may guarantee that the software is virus proof, or they may offer it in a convenient format (a subscription based service or a collection on CD-ROM).

Another trend away from public domain distribution is that authors have begun copyrighting their work to ensure that others do not make derivative works and market them as their own. Richard Stahlman's Free Software Foundation was created to develop an alternative to AT&T's UNIX software. Stahlman does copyright his programs but only to ensure that no one else will try to sell his software, and much of his software is considered as good or superior to commercial products.

The distinction between the market-based and public domain distribution systems is no longer as clear as it once was. With the development of shareware and copyrighted shareware, and the advent of third party publishers of public domain collections, there has been a gradual blurring of these two systems.

All of the distribution strategies discussed thus far entail some sort of publication. But there is a good deal of software which never goes beyond the individual or institution who developed it. Though some of this may be extremely useful, the incentives for publication are lacking. Developers may not perceive the rewards of broader dissemination as worth the effort required to debug and document their programs to make them available through clearinghouses or public domain libraries, and commercial publishers may not perceive

a market for the product. The library may choose to act like a publisher for some of these materials by collecting and/or cataloging them and making them available for use either locally or online. The library has a potentially large role to play as publisher of electronic resources created by itself or members of its community.

The Selection Process

Katherine Chiang (1987) expresses some reservations in describing the Mann Library's collection policy since the rapidly changing nature of software makes codification of a selection process difficult. But, as she points out, the lack of a policy can result in a collection of software which is wasteful and inefficient (p. 36). Despite the importance of a collection policy, of the seventy-four institutions responding to a 1985 ARL SPEC survey on microcomputer software policies, only 14 percent had a collection or circulation policy in place and another 20 percent were developing one. Only 5 percent had an institution-wide collection development policy (Association of Research Libraries, 1986, pp. 2-3). A survey done by *Choice* confirmed these results. The authors found few policies, and, where they existed, they were usually extensions of current policies rather than wholly separate documents (Dudley, 1986, p. 704).

In developing the selection process and collection policy, a library must decide whether to develop a single policy for all electronic resources or multiple policies for distinct types of resources, or choose to integrate this format into existing subject-oriented collection policies. Despite the variety of resources which one can appropriately consider as software, there is a tendency to lump these information products into one large category for the purposes of collection development.

The reader may argue for the usefulness of establishing a single set of selection criteria for all electronic resources. Yet as more and more materials are made available in digital format, the issue of form versus subject should become a nonissue. It is clear that, in certain cases, the electronic resource resembles print materials and may fit well into existing policies. This is true when dealing with data files, particularly primary or secondary information sources for which no print equivalent exists. In such cases, the overriding consideration will be the subject and scope of the resource. Even where electronic formats are to be integrated into existing policies, the policy authors must pay heed to the format's unique characteristics and address issues like system requirements.

Other categories of electronic resources may not integrate easily into existing policies. A survey of libraries who are acquiring bibliographic data tapes to load on local systems found that fewer

than 5 percent of the schools had collection development policies for electronic resources in place. Nevertheless, decisions about which indexes to purchase in print—CD-ROM or tape formats—are some of the most critical that are being made in libraries today and require policies which provide guidance in selecting format as well as subject.

Another area which clearly demands its own collection policy is software programs. Although one could argue whether educational software policies might more appropriately be incorporated into existing policies for particular subject areas, personal productivity and research tools do demand separate policies since these are functional (tools) rather than informational products and represent a new category of material for libraries.

Whether or not the library chooses to integrate computer files into existing policies or to develop a separate collection policy, elaborating selection criterion in a formal manner ensures that the library will undertake development of the collection in a coherent and systematic fashion. Examples of some academic library collection policies can be found in ARL SPEC Kit 123, *Microcomputer Software Policies in ARL Libraries* (Nollan, 1986).

Who are the Selectors?

In establishing collection policies for electronic resources, the library must take into consideration not only the selection criteria, but also who will be making the decisions. Where the collection development strategy is based on form rather than function/subject, the tendency may be to assign one person the responsibility for making all collection decisions involving software. Assignment of responsibility may well vary with the type of library and even within the same type of library. In small libraries, such as school libraries or small public libraries, all staff may contribute to decision-making (Dewey, 1984, p. 36). In some cases, administrators or reference staff may have selection responsibility or perhaps one staff who will be assigned and who has experience with microcomputers (Hannigan, 1985, p. 340). But in larger libraries, particularly in academic libraries, one of two models prevails—assignment of responsibility to collection development staff or assignment of responsibility to a microcomputer or software specialist. In the first model, attendant administrative and public service responsibilities usually reside elsewhere; in the second, the person who is responsible for selection is usually in a unit responsible for managing public access microcomputer services.

In libraries collecting software for teaching and research, requisite subject expertise in selecting programs as well as data files may mean that collection development is done by subject specialists. This is the case at the University of Florida (Beaubien, 1988, p. 667) as well

as at a number of other ARL libraries whose policies are included in this SPEC kit (Nollan, 1986).

On the other hand, at Cornell the need for computer expertise was the overriding factor in the Mann Library's decision to let all software selection rest with either the computer data files librarian or computer projects coordinator. It was hoped that within two or three years appropriate skills would be found among staff to facilitate moving the selection of computer files to subject specialists (Chiang & Curtis, 1987, p. 49). Many times these positions are newly created and these people often have not only collection responsibilities, but support and training responsibilities as well. In this model, faculty often play a major role in the selection of software (Dudley, 1986, p. 704) or subject experts are consulted or asked to make suggestions so that subject knowledge is available to the primary selector.

Carnegie Mellon created two positions, one responsible for software programs and one responsible for statistical and bibliographic data files. Eventually, these two functional responsibilities were reintegrated into other parts of the organization. This pattern of creating new positions and then integrating their functions into existing positions after a certain time allows the library to give new services the attention and visibility they need to be successful during the initial period of rapid growth. Those positions (lines) can then be used for other new services as needed.

As libraries gain greater familiarity with computer files, the responsibility for building collections of software will probably evolve to a hybrid model wherein subject experts may be responsible for some types of software (for example, courseware and subject data files) while a microcomputer specialist may select personal productivity software, and the head of reference or some committee of public service and systems staff may be responsible for the selection of bibliographic files.

Whether the final decision to purchase a single piece of software or data file resides with one person, or with different librarians throughout an institution, it is recommended that the collection policy be developed by a committee representing various interests (public service, systems, and collections) in the library. If responsibility for collection is to be dispersed among various staff members, Intner (1988) recommends some administrative coordination (perhaps by the head of collection development) to ensure a coherent collection and to minimize duplication of purchases (p. 13).

The main reason that libraries choose to create new positions for software or data file specialists is because this type of library service requires special skills. Hannigan (1985) surveyed libraries to identify priorities for staff development related to public access

microcomputing and found that, despite the fact that librarians may need new skills to enable them to select and evaluate software, few libraries are concerned with developing these skills in their staff. Hannigan addresses the critical need for staff development programs to develop competencies in selecting and evaluating software. She identified the following types of skills as necessary for selecting software: (1) fundamental skills in using microcomputers; (2) ability to recognize a program's objectives and match those with the library's objectives; (3) ability to read and analyze computer documentation; (4) ability to recognize errors and false information; (5) knowledge of a wide range of programs within a generic type (e.g., word processing programs); (6) recognition of the value of electronic communication and networks; and (7) ability to compare and contrast programs with similar objectives (p. 341). Johnson (1990) would add to this list an in-depth knowledge of the universe of computer files and how to locate them (p. 5).

Some of the skills which librarians need are technical, although the extent to which one requires computer expertise in the selection of materials will vary with the type of software and whether or not the selector must fully evaluate some of the software before purchasing it. The selector should at least have the knowledge to allow matching system requirements with available hardware. If the selector will also be supporting the software, then technical knowledge is critical. But developing good evaluative and critical thinking skills is equally important.

Rappaport (1985) suggests that the best way for librarians to develop computer skills and "discriminating, critical skills" is for the computer to become a routine part of the library and for staff to have time to "play with the computer." Once librarians begin using computing for their own productivity, they can take the next step into public access computing and help with the ongoing development of such services (p. 57).

Money, Money, Money

Software is expensive—not all software, but many of the popular packages are costly. Where will this money come from? Should a separate budget line be established? Budgeting will depend on whether a library chooses to have a single acquisition strategy for all electronic resources or whether collection policies are integrated into certain subject areas.

Casey Green, a researcher with USC's Center for Scholarly Information, has likened computing costs to a four legged stool encompassing hardware, software, support, and amortization/ maintenance. Most institutions have only focused on hardware since

this was the largest single cost. In fact, many academic libraries who received hardware as part of a grant may have overlooked the substantial costs of providing software and support. Rappaport (1985) notes that "a librarian ...told me they didn't even have any background information on the selection of their original collection of software because it came as a gift along with the hardware. Now they add software as they get extra money" (p. 57). Initial investment alone is insufficient if the library is to have a viable collection.

Because of the substantial initial investment, a library beginning to acquire software might want to use capital funds (equipment budget, for example) to purchase a start-up collection. Then additions to the collection and new versions of older software would be funded from operational costs or the materials/book budget. Another model used by the Academic Computing Department at Carnegie Mellon funds all new purchases from the capital budget and new versions and maintenance agreements/contracts out of operational/annual monies.

Software maintenance is a real cost that cannot be overlooked. Most hardware is obsolete within five years, and the software technology is changing even more quickly. A library may not need to have the latest in word processors, but neither does it want to provide tools which are considerably out of date. Any annual software budget will have to include funds for updates. Depending upon the scope of the collection, maintenance funds might be as much as 50 percent of the total budget. In fact, some software may be financially more like a serial than a book or other one-time purchase.

If the acquisition of software is going to be handled by subject specialists, then the library might do well to establish a separate budget line which can then be subdivided among the selectors. But such allocations require new algorithms for estimating costs. Unlike printed materials, software in the humanities and social sciences is not necessarily less expensive than its counterpart in the sciences. A good textual analysis program is likely to cost just as much or more than an equation analysis package.

One other possibility for funding purchases of electronic materials in academic libraries is cost sharing with academic departments or with the computing center where software under consideration has, respectively, a very narrow or very broad appeal. Cost sharing can entail considerable negotiation and politicking. In some cases, academic departments have already been collecting these materials for their faculty and students and may be reluctant to change the status quo at the risk of disenfranchising or inconveniencing faculty and students. Computer centers may wonder why the library is providing information in a format over which they have ultimate

control. The library has much to offer as a manager of these disparate resources but must avoid being perceived as usurping departmental authority. If the library takes the lead in coordinating purchase of these resources, not only is unnecessary duplication of resources avoided, but the library ensures that it remains a central source of information about what is available and how to obtain it (Johnson, 1990, p. 4).

Selection Criteria and Collection Policy

In developing selection criteria, the first thing to be defined is the scope of the collection. The scope or purpose of the collection is, to a great degree, dependent upon the type of library. For example, school libraries usually collect what could be termed courseware and probably also collect basic word processors and programming languages. Patrick Dewey (1984) lists several areas in which public libraries should collect software including computer literacy, programming languages, computer-assisted instruction, special interest programs like software for tax preparation or recipe doubling, library skills programs, personal productivity programs, software for children which can be useful for preschoolers as well as to supplement the local elementary school curriculum, and entertainment software (pp. 31-35).

The scope of the software collection in an academic library eludes such simple definition. In academic libraries, the reasons for collecting software programs are not always clear. Because software programs are functional (tools), libraries may have difficulty viewing them as an extension of the print collection—as a collection to support the research and teaching mission of the university—since the print collection is largely informational in nature. Libraries may feel most comfortable with a collection policy similar to Cornell's Mann Library which fits most closely with the library's traditional role as liaison between the information and the client. Cornell's collection is primarily one of data files and applications programs which are used to access, control, and analyze information (Chiang, 1987, p. 39). Other libraries may see no reason to limit the scope of the collection to traditional services.

The scope and purpose of the academic library's collection is generally not determined in a vacuum. Given the cost of these resources, the last thing the library wants is to duplicate a service being provided elsewhere on campus. Before embarking on a project to establish a software collection, every library needs to study the computing environment of the college and university. What role does the computing center play in support of public access microcomputing? What role do departments play? Are there any unserved or

underserved populations?

The public access microcomputer service or software collection is going to be shaped by the larger institutional environment. That is why Dudley (1986) found, in her survey for *Choice,* that software collections of academic libraries, some of which are traditional and some of which are completely new types of services, serve "a plurality of needs" (p. 704). In some institutions the library may be the only place providing software for public use, or its microcomputer facility may be one of a few facilities on campus. In these cases, the library may collect generic applications software like word processors. In other institutions, the computing center may fulfill this role, and the library's collection may be geared to meeting other needs.

While Dudley (1986) found that the majority of academic libraries are developing collections which contain mostly productivity software, some collections fulfill a reserve function and house only software used by faculty in direct support of their courses. Still others collect basic skills software or materials developed in house. Education libraries tend to collect K-12 software, often deposited by the producers. Despite the wide variety of models that the *Choice* survey found, few academic libraries are building a general instructional software collection to support both the curriculum and independent study—the central role of most academic print collections (p. 704).

There are a number of other issues that libraries must address in their collection policies that are unique to software. Should the library standardize on certain applications? Should the library purchase multiple copies of software? Should the library purchase only from selected publishers? Should the library acquire public domain software or locally developed software? Should the library maintain multiple versions of the software? What formats should the library collect? What hardware will the library support?

In determining the scope of the collection, the library must also decide whether to "standardize" on one or two packages in different application areas or to provide users with a variety of software. The former approach has the advantage of developing a considerable base of expertise but has the disadvantage of censoring the collection and dictating to the library patron which package best meets their needs.

Dewey (1984) states that variety is the overriding principle in selecting software for a public library collection—different kinds of software should address different ages, educational level, and interests. The goal of the collection should not be exhaustiveness but rather a selection of good programs. In some areas—such as entertainment—the library may want to collect a number of programs, in others—e.g., database management programs—one title may suffice

(pp. 30-31).

If a library chooses to standardize, it will probably also purchase multiple copies of the standard packages. If a library maintains a public computing facility which is used by its clients for word processing, then it must purchase multiple copies of the software. In other cases, the purchase of more than one copy of a program will be dictated by demand. If software, with the exception of courseware, is going to be used in teaching, then it is often possible to negotiate with producers for permission to use one copy for the entire class.

At the time that a library develops a collection policy, the staff should also consider whether the library will only collect commercially published software or also consider public domain or locally developed software. In order to provide a greater variety of software, many public and academic libraries acquire public domain collections. Purchasing public domain software also allows the library to freely circulate the software without fear of piracy. A library may also decide that one of its functions should be as an archive of locally developed software which faculty have created to support their research or teaching.

A library must determine if it will maintain old versions of software it collects. While libraries don't usually discard earlier versions of printed materials, taking on an archival role for even locally developed materials may be beyond the capability or desires of the library. Although most program files are upwardly compatible, data files are not. That is, a report or article might be based on a data file that was subsequently revised. If all libraries chose to discard that file, then no record of the researcher's primary data would exist. It is critical that both individual libraries and the profession address this issue sooner rather than later. (In 1990, Columbia University's School of Library and Information Science sponsored a brainstorming session on the development of a microcomputer software archive. Although, the primary purpose of the archive was viewed by most participants as historical in nature, there was strong interest voiced in some type of comprehensive archive which could be used for patent research and would ensure that programs existed somewhere which would be able to read data files that current software was unable to read.)

Libraries must also determine the formats which they will collect. Many libraries only support microcomputer formats, while others may collect materials which will physically reside outside the libraries (e.g., data tapes may be housed in the computer center). Libraries may want to consider to what extent they will be responsible for collecting online data or software outside the physical library.

Libraries can provide access to this material through telecommunications links, cataloging, and some level of support.

In addition to determining the scope of the collection and the type of materials to be collected, the library will want to determine what hardware and operating systems it will support. In an ideal world, a library should make decisions about what software to acquire independent of and before any hardware decisions are made. The library would assess the needs of its clients, find suitable software, and then purchase the hardware to support it. But few libraries, if any, let software decisions drive hardware purchases; in almost every instance the reverse is true. If the software will be used outside the library, the library's own machines become less important in decision-making, though the library may still choose to support only those hardware environments which they own so that making back-up copies, providing support, or evaluating products is feasible. If the library is part of a larger organization (university, school district, or corporation) it will want to collect software which is compatible with the hardware environment supported by the host institution.

Evaluation of Computer Files

The question of what constitutes an appropriate evaluation of software is dependent upon the type of software being collected and the purpose to which it will be applied. Evaluation can be accomplished in various ways. Libraries can acquire software which has a thirty day return and perform a complete hands-on evaluation. School librarians often have access to regional evaluation centers. Software is regularly exhibited at trade shows affiliated with conferences, such as EDUCOM, MacWorld, NECC, though little is marketed at ALA. Local computer stores are often useful for evaluating generic applications software. There are published reviews and, last but not least, librarians can talk to users of the software or local user groups or other experts (Dewey, 1988, pp. 66-67).

The time spent on evaluation must justify the use and cost of the software. If the library is making institution-wide decisions and recommending the "standard software" to be used throughout the institution, then hands-on evaluation and considerable testing of competing products is useful. An expensive software package may seem to require substantial hands-on evaluation while less expensive products may merit less attention.

Sometimes reading published reviews or descriptions may be sufficient. However, most reviews are not written with a library audience in mind. Patrick Dewey's (1988) chapter in *The Library Microcomputer Environment* provides a useful review of published software reviews. The article covers the anatomy of a review and

gives some caveats about the review literature. It is clear from Dewey's article that selecting software is not yet as simple as selecting a book, but with a certain amount of skepticism and a grain or two of salt, librarians can rely upon published reviews.

At some point, when there is no published information about a software package or information about how the library intends to use it, a library may need to undertake software evaluation. There is a huge volume of material published on software evaluation particularly as it pertains to educational software. There are many published "guides" or forms to help professional evaluators and software purchasers evaluate software. Most guides are geared towards evaluating software used in an educational context. Jones and Vaughan's 1983 publication, *Guide to Guides*, and a recent article by Schueckeler and Shuell (1989), provide a comparative survey of some of these guides and evaluation forms. Since most of the evaluation guidelines are geared to K-12 educational software in the classroom environment, several librarians have developed short lists of criteria which may be more appropriate to the library environment. Clyde and Joyce (1985) present three sets of evaluation criteria for the school library: (1) general, (2) criteria for library management software, and (3) criteria for educational software. Troutner (1989) gives guidelines for evaluating three types of educational software. Cargill (1987) provides guidelines for different types of productivity software (word processing, communications, and spreadsheets) as well as some general criteria. Dewey (1984) also provides a useful list for the public librarian who may be purchasing productivity, entertainment, or educational software.

The criteria established by these authors are unique to computer files and to programs in particular. Margaret Johnson (1990) takes a different approach when she analyzes eight general criteria used at Columbia in making collection decisions regardless of format and applies them to computer files. These eight are: (1) relevance of material to program needs, (2) scope of treatment, (3) ability to fill existing gaps, (4) quality of scholarship, (5) currency of information, (6) accessibility of information, (7) language, and (8) cost (pp. 7-9). Johnson's article is particularly useful since it does not focus exclusively on programs, but can be applied more generally to all types of computer files.

None of these lists, with the possible exception of the Clyde and Joyce article, are definitive, but they may be useful as a starting point. One set of evaluation criteria for one type of software cannot always be easily transferred to another type of software. Chiang and Curtis (1987) suggest that one might develop one's own guidelines for certain genres or subject areas of software. But this task is by

no means simple, and it requires a high degree of familiarity with the genre. Sometimes comparative reviews give a description of what a program should do and how programs measure up to that standard. If no descriptive standard exists, the selector might develop his/her own by examining individual reviews. But if no published reviews exist, the selector will have to create their own standard (p. 54). Such standards are only useful for comparative reviews. Many software programs, especially those used in higher education, are unique. They often derive from a faculty member's syllabus and are as personal as lecture notes, or they may fulfill a need which had hitherto gone unmet. Applying general collection guidelines, such as the Columbia criteria, may be particularly useful in such situations.

The evaluation of any program can be divided into four areas: technical evaluation, design, content, and support. Librarians should concern themselves with the following aspects of each of these.

Technical Concerns

Technical evaluation encompasses hardware and operating system compatibility and the reliability of the software (is it bug free, does it run efficiently, does it work as advertised?). If software is going to be used in the library only, then hardware compatibility is critical. As hardware has become more complex, the system requirements for any individual piece of software must be carefully scrutinized. Obviously, machine compatibility is critical, but software may also require certain graphics functionality (a particular card or monitor); it may require so much internal memory or so much disk space. Data files may only be compatible with certain data formats or software programs. Few personal productivity packages can be run from a floppy disk anymore; they must be loaded onto a hard disk and this may have implications for machine maintenance and licensing. A library may need to determine if software can be placed on a network (legally and technically) so that multiple users can have simultaneous access.

Libraries will want to collect software and data files for which there is some quality insurance—though this is not always possible. Just as libraries judge the quality of print materials by their publisher, software selectors should be able to do the same for electronic resources. However, even commercial programs are regularly released with bugs to say nothing about public domain software. Viruses which are introduced intentionally into software are also cause for concern. Although there has been, at least, one scandal wherein a commercial software package was released with a virus, the most susceptible sources are public domain bulletin boards. Many user groups and software libraries such as Compuserve's and Boston

Computer Society guarantee programs which are virus free, but then the user must pay for this benefit. Data files should be evaluated for data integrity and validity. The problem of quality assurance is even more difficult with unpublished software. Some clearinghouses which distribute this type of software provide no guarantee of quality—either technical or content; others have some peer review process. The InterUniversity Consortium for Political and Social Research (ICPSR) ranks its data files from one to four based on their quality, but the ICPSR is the exception rather than the rule. Since the integrity of data can be so easily compromised in the electronic environment, librarians will be called upon to become adept in recognizing reliable and quality sources of computer files.

The type of software determines the other specific characteristics that should be tested. Performance evaluation and benchmark testing are common with generic applications software. The speed of sorting or maximum size of a record are critical in evaluating database management systems but irrelevant to evaluating a drill-and-practice program.

Software Design

Software design encompasses areas such as screen design, menu design/branching, integration of graphics or sound, how effectively the software makes use of hardware, and how it interacts with the user. Graphics are evaluated for aesthetic value, appropriateness, and technical quality; textual materials must be examined for accuracy, readability, and motivational quality. The program needs to be examined for ease of use or ease of learning for its intended audience and how these factors are balanced by functionality. To those practiced in evaluating other media (particularly school librarians and media specialists), much of the design evaluation process will seem familiar, but the interplay among the components is new (Swigger, 1986, p. 285).

Content

The content of a program should be evaluated by what the program does and how effectively it meets the user's objectives. Johnson (1990) notes that determining the relevance of a computer file is a skill new to most selectors and bibliographers. It requires familiarity with the published and unpublished "universe of computer files." In the academic environment, it may also mean knowing what faculty and students are already using, what has been locally developed, or what has been acquired by departments (p. 7).

It also means that the selector must know how the computer file will be used. While it may be fairly straightforward to assess

the content of a data file, a word processor, or numerical analysis program in certain contexts (such as in business), many more questions are raised when that word processor, numerical analysis program, or data file are to be used in educational contexts. It is difficult to anticipate how a particular piece of software may be used. While one person may use a spreadsheet to do budget analysis, another one may use it to create tables to insert into a text document, and another may use it in classes to illustrate certain civil engineering problems. Because a single piece of software can be used in so many ways, one cannot rely on product summaries to determine content. Research in areas as widely disparate as discourse analysis and information retrieval have not yet provided reliable techniques for determining content (Swigger, 1986, p. 288).

The evaluation of information technologies and their impact on learning is a continuing source of debate among those who create and use computing in teaching. As Swigger points out, the problem with many of the current evaluation guidelines is that "they assume the important characteristics of [software] in terms of learning consequences are known" (p. 291).

Content of courseware and games should also be examined for offensive material. Issues dealing with equity oriented guidelines for the selection of software are addressed by Patricia Campbell (1986).

Support

In addition to the software's characteristics, one should also evaluate the quality of documentation (online and printed); tutorials, and other supporting products to help in learning a program, and vendor support services such as hotlines, update policies, and costs; warranties and return policies. Last, the benefits and quality of a data file or program must be weighed against local support issues—that is, the financial, personnel, and physical constraints of the library.

Sources of Software Information

It is frequently difficult to find even the most basic level of information about some software. That is because software eludes traditional bibliographic control mechanisms which help libraries identify and gain access to materials. One reason that software defies control is because the avenues of distribution are many and they include many avenues outside the normal publishing infrastructure (see the section entitled Mode of Publication/Distribution). Second, the software publishing industry is highly volatile.

In the year following the publication of the *Directory of Software Sources for Higher Education,* it is estimated that 50 percent of the commercial sources of software listed were no longer publishing and/

or distributing software. The reasons for this turnover are many. Much commercial software production is a cottage industry with small mom and pop shops dotting the countryside, and these "family businesses" are born and die with amazing rapidity. Many traditional publishing houses who entered into the software publishing industry at the onset of the microcomputer revolution eventually dropped out for lack of market share. The late 1980s witnessed a major shakedown in the software publishing industry as a whole with many smaller entrepreneurial shops being taken over by larger software companies.

In part, due to this volatility, the efforts to provide efficient indexes to this body of material has been equally scattered and volatile. While the mid-1980s saw an explosion in the number and variety of sources which sought to describe and evaluate software, many of these efforts were one time publications that were soon out of date. (In the last two years, both the *ICP Directory of Software,* one of the most comprehensive directories of mainframe and minicomputer software, and Elsevier's *Software Catalog* ceased publication. MENU Publishing, who maintains the database from which the *Software Catalog* is derived, says that they will continue maintaining the database for the time being. *Software Reports,* an evaluation service for K-12 software, has also ceased publication.) There are some standard sources emerging for the more common types of applications software, but many other types of computer files remain undocumented. As the body of electronically published data seems to increase logarithmically, the problems of bibliographic control of this material reaches crisis proportions.

Types of Software Information

The software selector may require up to three levels of information about a particular program in order to make an informed decision. At the most basic level the selector needs to be able to locate a program to match his client's needs. At this level, he/she may only need some sort of description of the program. This description must include a complete statement of system requirements as well as a description of the program's capabilities or scope. If several programs exist, the selector may need to find out which is the best program. One requires evaluative information to be able to make this decision. But even when not selecting among programs, the detail of a good evaluation may be necessary to decide whether the software should be acquired. The type and depth of printed evaluations can differ substantially from source to source. For example, some sources of software are selected lists of the best software in a particular area (e.g., the EDUCOM/NCRIPTAL Awards catalog), but the actual evaluative information that they publish is minimal. Others simply

rate programs, and some sources contain in-depth evaluations. As Dewey (1988) points out, spoken reviews (verbal or electronic) from users of the program or user groups are particularly helpful (p. 67). Lastly, the selector may need use information to understand how the software might be applied to a user's needs. This anecdotal information can help address uses of software that the publisher may not have anticipated.

The amount and type of information to be found about any one type of software varies significantly. While personal productivity tools, CAI for K-12, and commercial bibliographic data files are relatively well indexed, and there is a significant amount of evaluative information about the commercially published materials, information about certain data files or research software or software for higher education can be more difficult, if not impossible, to find. So much of this information is still communicated informally—word of mouth, conferences, or online bulletin boards.

Resources

The number and variety of software information sources is overwhelming (see Appendix). Many different directories exist to help the selector do retrospective collection development and hundreds of journals, magazines, newsletters, and electronic bulletin boards exist to help keep selectors current. Some of the sources are horizontal, covering broad genres of software or software for particular machines, and some are vertical, covering specific subject areas.

In order to help software selectors find some of the more obscure indexes to software, librarians have written several guides to software resources. The most recent and most comprehensive is Carol Truett's *Microcomputer Software Sources.* This unique source contains eight sections which provide descriptive and/or evaluative annotations on comprehensive software sources, business software sources (this section is divided into eleven chapters covering different vertical markets such as agriculture, law, medicine, retail trade, design and engineering, and science and technology), educational software sources, library applications, machine-specific software sources, and public domain software. It also has a section which serves as an introduction to the literature of computing.

The author's *Directory of Software Sources for Higher Education* is a similar resource listing commercial and noncommercial sources for software programs and data files developed specifically for research and teaching in higher education, as well as catalogs and databases of software, published bibliographies of software, journals, review sources, and organizations which are either developing software or collecting information about software. Though the focus of this

directory is information about software used in higher education, the scope of these sources is broad enough to be useful to any library. *SoftInfo,* a database sponsored by IBM, developed by the University of Bridgeport, and available through the ISAAC system at the University of Washington (bitnet and internet accessible), provides information on a wide variety of resources (databases, journals, print catalogs, and organizations) which are sources for information on IBM software. Truett's book should be the first stop for anyone looking for software programs in an area with which they are unfamiliar and should be in every library which collects software.

Software Programs

Equally important are a number of directories purporting to be the "Books in Print" for software. Some focus exclusively on microcomputer software such as Bowker's *Software Encyclopedia* or the *Datapro Directory of Microcomputer Software.* Elsevier's *Software Catalog* (also published on Dialog), is available in six editions: microcomputers, minicomputers, business software, science and engineering, health professions, and systems software. Datapro does publish a software directory covering larger machines, but the cost may be prohibitive to smaller libraries. Those seeking software for larger machines may wish to consult directories published by hardware vendors.

There are many directories which only cover software for certain machines. DEC has a catalog for third party commercial software which runs on its Ultrix operating system; Sun's *Catalyst* catalog lists third party software research and education. Most other mainframe and minicomputer/workstation vendors have similar catalogs of software.

Much software for larger machines and for research is developed by federal agencies and their contractors, and there are several directories of software available from the government including the *Directory of Computer Software* from NTIS and the *COSMIC Catalog* which lists software from the COSMIC clearinghouse at the University of Georgia, a distributor of programs from NASA. Other directories are limited to certain types of software such as *T.E.S.S., The Educational Software Selector,* or to even more specialized subject areas (e.g., the CALICO database which lists software for computer-assisted language learning). The problem with many smaller directories is that they are updated infrequently if at all. There are a significant number of sources in Truett's directory which have not been published since 1984.

There are hundreds of journals to help the software selector keep abreast of developments in the software market, not only through

product announcements and reviews but through ads. Some of these are general such as *Byte* and *InfoWorld,* but many are specific to certain machines, such as *PC Magazine* or *PC World* for IBM and compatibles software, *MacUser* and *MacWorld* for Macintosh software, *A+* or *InCider* for Apples, *Ahoy* for Commodores, and most recently *NeXTWorld* for NeXT machines. Subject specialists doing software selection may want to scan the advertisements and product announcements in computing publications in their particular area, such as *CIME: Computers in Mechanical Engineering,* or in subject specific journals covering technology, such as *Analytical Chemistry* or *Civil Engineering.*

Data Files

There is a dearth of information about data files with the exception of bibliographic data files. The most comprehensive sources are *Computer-readable Databases* and *The Data Base Directory.* Cuadra Associates, who has published its *Directory of Online Data Bases* since 1979, recently began issuing the *Directory of Portable Databases* which covers CD-ROM products. The standard source for CD-ROMs is Meckler's *CD-ROMs in Print.* Updates appear in *CD-ROM Librarian.* Sources of government data files include *A Directory of Computerized Data Files: A Guide to U.S. Government Information in Machine-Readable Form* from NTIS, *Federal Statistical Data Bases* from Oryx, the *Federal Data Base Finder* and the *State Data and Database Finder,* both from Information USA. Files produced by individuals elude any sort of bibliographic control unless they are deposited with a clearinghouse like the Interuniversity Consortium for Political and Social Research.

Since many electronic resources are becoming available over Internet and other national networks, there have been some efforts made to catalog these resources. Both *Internet Resource Guide* and Art St. George's *Catalog of Internet Accessible Library Catalogs and Databases* are limited but useful introductions to the scope of resources available on that network.

Public Domain Directories

Sources of information about public domain software are less than current. Most of the directories listed in Truett's book and in Fred Sissine's 1988 article were published prior to 1984. Glossbrenner's *Master Guide to Free Software for IBMs and Compatible Computers,* 1989, does include a core collection of software, but most importantly tells users how to locate this software. Once users are familiar with how to access the major public domain archives, lists of the archive's current contents can be obtained. Glossbrenner's 1984 book, though

out of date, provides similar information for "every brand of personal and home computer." Some user groups, such as the PC-SIG group and Boston Computer Society, publish print catalogs of their software archives.

Evaluative Information

There are a few compendia of software reviews which attempt to be comprehensive in scope; most are quite expensive. Half of the *Datapro Reports on Microcomputers,* a looseleaf service, is devoted to software. These are comparative reviews which include overview charts of all software packages examined and in-depth evaluations (twelve to fourteen pages in length) of selected packages and are limited to courseware. *Software Digest Ratings Report* and the *Software Digest Macintosh Ratings Report* are published ten times per year and cover one type of package per issue. *Software Reviews on File* is like *Book Review Digest.* For each package covered, it includes the publisher's description and one or more excerpts from published articles that review the product.

Software for K-12 has benefited from a well-developed curricular evaluation process. Consortia and various state and regional clearinghouses have been established to review courseware. While some of the sources attempt to be comprehensive, such as *EPIE PRO/FILES,* others only list a select number of programs in an effort to help guide teachers and librarians to the best applications like *Only the Best: The Discriminating Software Guide for Preschool-Grade 12* and the *Educational Software Preview Guide.* Comprehensive evaluation sources include *RICE: Resources in Computer Education,* a database compiled by the Northwest Regional Education Laboratory from its MicroSIFT evaluations, EPIE Micro-Courseware PRO/FILES, an annual publication with bi-monthly updates that provides comprehensive evaluations of over 500 packages. (MicroSIFT no longer publishes individual reviews of software, but NWREL continues its evaluation activities through publication of Courseware Evaluation Reports. These reports are an analysis and synthesis of available evaluation data that focus on specific areas of microcomputer use. These reports are added to the RICE database.)

The best sources for reviews of software, other than K-12 programs, are to be found in the journal literature. Many of the machine specific journals, like *MacWorld* or *PC World,* will do extensive product comparisons. As noted earlier, there are many discipline specific computing journals which include software reviews. Furthermore, many professional, scholarly, and teaching journals are beginning to include software reviews just as they include book reviews. For example, *French Review* and *Hispania* carry reviews

of French and Spanish software respectively and the *Journal of Nutrition Education* covers nutrition software. *Choice* is a useful source of reviews of not only software programs appropriate for academic libraries, but also of data files and online databases. The reviews in *Booklist,* like *Choice,* are from a library perspective, however neither source does a comprehensive review. The focus of the reviews is more on content than on technical aspects. *Library Software Review* provides more extensive reviews, but most of the reviews are geared toward library applications rather than public access applications. Again, one will find that any evaluative information on nonbibliographic data files is virtually nonexistent.

To locate these reviews, selectors can use one of several online databases or printed indexes which cover computing literature including the *Microcomputer Index,* the *Computer Database,* or *The Computer Literature Index. ERIC* or *CIJE* cover all of the teaching journals and some of the scholarly journals which publish software reviews. Selectors should also search relevant databases which cover literature for specific subject areas like *Compendex/Engineering Index.*

Software Use Information

Unfortunately, it is impossible to find any sort of directory of how people are using software. Vendors sometimes attempt to gather this type of information. For instance, Apple Higher Education has been developing a database of uses of Macintosh technology in higher education which does include how various Macintosh third party applications are being used, and companies like Ashton-Tate, Microsoft, and Addison-Wesley include anecdotes in their newsletters concerning how people are using their software. One can also search for such information in "how we did it good" type articles in the education and library literature. But most of use information is communicated much more informally—by word of mouth, face to face, or via electronic bulletin boards. Local user groups may be helpful in locating someone familiar with a particular application, and vendors sometimes will provide referrals.

The Limits of Printed Finding Tools

The software selector may quickly reach the limits of usefulness of printed sources. Knowledge of unpublished files is usually limited to their creators and other insiders. How does the librarian become an insider? Johnson (1990) suggests reading the literature of the discipline for references to computer files, networking not only with other librarians who collect computer files but also with other professionals through discipline-based scholarly associations, and

outreach to faculty who produce or use these files (p. 6). The electronic environment with access to online bulletin boards such as PACS-L (public access catalogs) or the Humanist ListServ (applications of computers in the humanities) can facilitate such networking and information exchange.

Acquiring Software

Acquisition of software often requires different procedures than acquisition of other types of library materials. It requires the acquisition's staff to find sources of the requested computer file and then find the best price. Though a certain percentage of published software programs may be procured through library jobbers, jobbers handle only a small percentage of available software titles. The library may be forced to deal directly with producers of computer files or other distributors, and often this entails some degree of negotiation. There is little need for negotiated agreements for off-the-shelf software programs and CD-ROM databases when one is purchasing single copies. However, the library may find itself negotiating software purchase agreements with vendors when purchasing multiple copies, developing institution-wide purchase agreements, or seeking permissions not granted in the license (e.g., classroom use or network use).

Even when the existing terms of the license and the price are acceptable to the library, the library may find it advantageous to purchase from the software publisher rather than some other distributor. Licenses will often sound much more restrictive than the publisher's intent. A library may choose to negotiate new terms with a publisher, and, in many cases, publishers are eager to comply. As such activity can be time-consuming, many libraries have adopted a policy, recommended by ALA counsel, whereby every purchase order carries some statement to the effect that the software would be lent in the manner of other library materials. This statement was considered to be binding if the publisher did not object. However, if the library purchases software from a jobber or discount house, such statements would be void since the publisher never had an opportunity to agree to them. If the library is required to sign a license agreement in order to ensure warranty rights and support, prior to signing, the librarian should confer with counsel to determine any municipal, university, corporate, or school district policies in this area.

The host organization often has its own policies concerning the purchase of software owing to the cost. The library may be asked to follow a procedure different than one it uses for purchasing other library materials when buying software. Whereas library materials

may not be subject to the scrutiny of purchasing departments, software purchases often are, especially if they are over a certain dollar amount. It behooves the library to try to integrate the acquisition process for software into the existing process for other materials in order to expedite purchase and have the freedom to purchase what it desires.

Another problem the library is likely to face is in terms of payment. Many software vendors are still unaware of the library as a potential market for their products, and they do not understand how library acquisitions function. Though they may have the means for dealing with institutional accounts, they are better equipped to deal with individual consumers with credit card in hand (Chiang & Curtis, 1987, p. 57). It is hoped that over time the situation will begin to change. But judging from the insignificant number of software vendors at library conferences, it is unlikely that vendors are actively courting this market and will be responsive to its needs.

Discounts

There are a number of ways to stretch the software dollar. For stand alone software, most discounts are based on the number of copies purchased. Site licenses and multiple copy discounts for stand-alone software have evolved as a way to provide wider and more inexpensive access to software. True site licenses where one pays one price for the privilege of unlimited copying is rare, in part because vendors have little control over who makes the copies and whether they are used off site. Usually the vendor will offer big discounts for multiple copies. Sometimes in exchange for a substantial discount, the purchaser has the responsibility of copying disks and manuals. Sometimes there are discounts based on laboratory packages. These packages usually include one complete copy of software and documentation and some additional number of software programs.

Owing to the proliferation in local area networks, many vendors have developed network licensing terms. It will usually cost an institution more to make software available over a network. Network licenses can be based on the potential number of users (size of the population served), actual number of users (how many will actually use a particular numeric modeling package), or the number of ports or terminals with access to the software on the network.

Even when the library is purchasing single copies of software there are discount options available. The library should first investigate whether there are any institutional, regional, or consortia agreements with a particular vendor in which the library can participate. The library may choose to purchase demo copies or crippled copies of software that may be suitable for teaching but not for actual production. A program may be crippled by disallowing

certain functions such as the ability to save a program or print. The library can also opt to purchase through a software discount house. The computing magazines are filled with ads for discount houses like Mac Connection through which end-users can get substantial discounts on software.

Academic and school libraries can often benefit from educational discounts, though these are usually only available on noneducational products. These discounts can be substantial. Companies like Microsoft, Borland, or Claris separately package software for educational institutions and often offer a tiered structure of discounts based on the amount an institution may spend on their products in a given year.

Others may simply discount the product for any educational institution. In some of these cases the discounts are standard, but in other cases the library may have to negotiate with individual vendors. There are also software discount houses (brokers) who specialize in products for education. These organizations have already negotiated with vendors for discounts which may be based on quantity.

Evaluating the Collection

The software collection needs to be continually evaluated and assessed. Are there enough titles to satisfy demand? Enough copies? Are there gaps in the collection? Is there enough variety in the types of hardware supported? The level of expertise? What is the condition of the collection? Does it need bolstering or weeding? (Intner, 1988, p. 16). It seems strange to talk about weeding a collection of materials which are less than ten years old, but software quickly becomes obsolete. One question the library needs to ask is the extent to which the collection should be archival as well as current. If the library needs to maintain an archival collection, then it will probably need to maintain the hardware on which to run the software. This may become increasingly difficult if the hardware (e.g., the DEC Rainbow) is no longer manufactured and parts are difficult to find. At some point the library will probably want to let go of trailing edge technology and make a determination that it won't continue to collect for certain kinds of hardware.

It is equally important to assess how the scope of the collection is matching user needs. Both Intner (1988, p. 17) and Polly (1985, p. 158) stress the importance of ongoing consultation with end-users. Often users know more about what is the next bestseller in business or entertainment software before the librarian and they can help the librarian anticipate demand for certain software. As Intner says, the library cannot be faulted for not forecasting exactly what its users'

needs will be, but it can be faulted for not adjusting collection policy and practice to meet those needs (Intner, 1988, p. 17).

Conclusion

The selection and acquisition process for computer files remains a complex task a decade after these materials were first introduced into libraries, and selection is unlikely to become easier as more and more material becomes available in digital format. The library must seek out the most appropriate models for collection development to enable it to successfully implement a public access microcomputing service. Some of these models will be based on, and grow out of, existing library models, but many will be new. The greatest challenge awaits the selectors who must seek out, from among the world of published and unpublished files, those which anticipate and meet their patron needs and develop the expertise to enable them to make sound judgments about the value of each piece of software.

APPENDIX

Resources

These resources were either selected because they are comprehensive directories for a particular type of computer file or are good examples of some of the many types of resources available to help locate and evaluate software. Journals mentioned in that section are not included in this list.

Directories of Directories

Seiden, P. (Ed.). (1987). *Directory of software sources for higher education: A guide to instructional applications.* Princeton, NJ: Peterson's Guides.

SoftInfo Database. Available through the ISAAC (Information System for Advanced Academic Computing). ISAAC, m/s FC-06, University of Washington, Seattle, WA 98195, 206/543-5604, Bitnet: *ISAAC@UWAEE.BITNET*

Truett, C. (1990). *Microcomputer software sources: A guide for buyers, librarians, programmers, business people, and educators.* Englewood, CO: Libraries Unlimited.

Directories of Computer Programs—Comprehensive

Datapro Research Corporation. (1980). *Datapro directory of microcomputer software.* Delran, NJ: Datapro Research Corporation
(2 base volumes plus monthly issues).

Software catalog microcomputers. New York: Elsevier Science Publishing Company, Inc.
Spring 1990 will be the last printed edition. Menu Publishing says that they will continue to update their database. For now, however, it is unlikely that they will continue developing the database in the long term future. Available on the DIALOG Information Retrieval Service, file 263 as The Software Database.

The Software encyclopedia. (1985) New York: R.R. Bowker.
Available on the DIALOG Information Retrieval Service as Microcomputer Software and Hardware Guide, File 278.

Directories of Computer Programs—For specific machines

Catalyst: A catalog of international third-party hardware and software for the education/research community. Mountain View, CA: Sun Microsystems, Inc.

Digital Equipment Corporation. *Ultrix software sourcebook including both VAX and RISC based applications,* 5th ed.
Available from: Digital Equipment Corporation, Media Fulfillment, NR02-I/J5, 444 Whitney St., P.O. Box 5000, Northboro, MA 01532-9976.

Directories of Computer Programs—Specialized

U.S. Department of Commerce. National Technical Information Service. (1980). *Directory of computer software.* Springfield, VA: National Technical Information Service.
Annual since 1980. Software created by federal agencies and their contractors for mainframes, mini and microcomputers.

COSMIC Software Catalog. Athens, GA: University of Georgia.
Catalog to programs in COSMIC, the sole distributor of NASA developed computer software.

CALICO Database. CALICO, 3078 JKHB, Brigham Young University, Provo, UT 84602, 801/378-6533.
A conclusive and complete reference service to information relating to any application of high technology to language. Database includes a software directory as well as a human resource directory and bibliographic database.

TESS, the Educational Software Selector/EPIE Institute. (1986). New York: Teacher's College Press.
1986-87 version and 1988 supplement with 1991 edition forthcoming. Most comprehensive directory of educational software for preschool through college.

Directories of Data Files

CD-ROMs in print: An international guide. (1988). Westport, CT: Meckler.
Updated by the CD-ROM Librarian. A directory of 240 optical based products and their producers.

Moore, L. G. (Ed.). (1984). *DataBase directory, 1984-85.* (1984-). White Plains, NY: Knowledge Industry Publications and American Society for Information Science.
Available through BRS, file KIPD. Contains information on 2,650 databases accessible online in North America. Covers full text, textual and numeric, numeric, property, bibliographic and referral type files.

Directory of computerized data files and related technical reports: A guide to U.S. government information in machine-readable form. (1980-). Springfield, VA: U.S. Dept. of Commerce, NTIS.
Lists more than 2,500 numeric and textual data files from some fifty federal agencies.

Directory of online databases. New York: Cuadra/Elsevier.
2 issues and 2 supplements/year. Contains information on over 4,600 bibliographic, numeric, full-text, referral, and software databases available through an online service, over networks, leased lines, or dial up to the public or to organizations.

Directory of portable databases.(1990-). New York: Cuadra/Elsevier.
2 issues/year. Lists 583 products available on CD-ROM, diskette, or magnetic tape. Covers the following types of files: audio, bibliographic, full-text, full-text/image, image, numeric, referral, software, and textual/numeric.

Zarozny, S. (Ed.). *Federal data base finder: A directory of free and fee-based data bases and files available from the federal government,* (2d ed.) (1984-85). Chevy Chase, MD: Information USA, Inc.

Evinger, W. R. (Comp.). (1988). *Federal statistical data bases: A comprehensive catalog of current machine-readable and online files.* Phoenix, AZ: Oryx.
Successor to the NTIS Directory of Federal Statistical Data Files. Covers 1,200 files most available on magnetic tape rather than online or on diskette.

Guide to resources and services: Inter-university Consortium for Political and Social Research. Ann Arbor, MI: ICPSR.
Lists over 25,000 files in approximately 1,600 titles held by ICPSR.

Marcaccio, K. Y. (Ed.). (1989). *Computer-readable databases: A directory and data sourcebook,* 5th ed. Detroit, MI: Gale Research, Inc.
Available on Dialog, File 230. Includes information on 4,200 databases which are available online or in CD-ROM formats.

Lesko, M. (1989). *State data and database finder.* (1989-). Chevy Chase, MD: Information USA.

Directories of Network Accessible Online Resources

Internet resource guide. (1989). Cambridge, MA: NSF Network Service Center.
The resource guide has information on computational resources, library catalogs, software archives, white pages, networks, and network information centers. To obtain information about receiving the guide, contact the NSF Network Service Center at 617/873-3400 or send electronic mail to *resource-guide@nnsc.nsf.net*

St. George, A., & Larsen, R. *Internet-accessible library catalogs and databases.*
The directory contains a listing of over 100 resources including library catalogs and databases, dial-up libraries, campuswide online information systems, and bulletin board systems. Information on accessing the catalog may be obtained by sending electronic mail to *stgeorge@unmb.bitnet* or *stgeorge@bootes.unm.edu*

Sources of Public Domain Software Information

Glossbrenner, A. (1988). *Alfred Glossbrenner's master guide to free software for IBMs and compatible computers.* New York: St. Martin's Press.
The book is in three sections: an introduction to software basics covering bulletin boards, shareware organizations, clubs and other outlets for software; descriptions of specific online systems, catalogs, and other sources; and recommendations for specific programs.

Glossbrenner, A. (1984). *How to get free software: The master guide to free programs for every brand of personal or home computer.* New York: St. Martin's Press.

Although somewhat out of date, the basic information in this volume is still very valuable as an introduction to procedures for finding and acquiring public domain software.

Evaluation Sources—General

Datapro reports on microcomputers. Delran, NJ: Datapro Research, Inc.
Looseleaf service. Comparative reviews of business software with in-depth evaluations.

Software digest. Philadelphia, PA: Software Digest, Inc.
Monthly. Tests IBM PC software by focusing on one group of competing programs (word processing, spreadsheets, desktop publishing) at a time. Programs are tested for performance, error handling, versatility, ease of learning, and ease of use. Software digest now publishes a similar report for Macintosh software.

Software reviews on file. New York: Facts on File.
Monthly. Contains digests of reviews from a variety of journals for business, utility, educational, games, and personal software. Each issue contains fifty short reviews.

Evaluation Sources—Educational

Educational software preview guide. Menlo Park, CA: Educational Software Evaluation Consortium.
Annual. Available through ERIC. Lists commercially available instructional programs for students in K-12, that have been favorably reviewed by members of the Educational Software Evaluation Consortium.

EPIE micro-courseware PRO/FILES and evaluations.
Available from EPIE, P.O. Box 620, Stony Brook, NY 11790, 212/678-3340. Full product evaluations done by EPIE (Educational Products Information Exchange) Institute, the Consumers Union, Columbia's Microcomputer Resource Center, and six school districts.

RICE: Resources in Computer Education. Northwest Regional Educational Laboratory.
Available through BRS. The NWREL's comprehensive database of descriptive and evaluative information about microcomputer software for K-12 compiled through the MicroSIFT program. MicroSIFT ceased publishing individual evaluations in 1985 but continues to produce special reports on classes of software. These reports are indexed in RICE.

Only the best: The discriminating software guide for preschool-grade 12. Sacramento, CA: Education News Service.
Annual guide lists over 200 programs chosen from 8,000 evaluations conducted by thirty-two evaluation services.

Indexes/Abstracts to Evaluations Appearing in Journals

Computer database. Belmont, CA: Information Access Corporation.
Available through Dialog Information Retrieval Services, File 275, on CD-ROM or magnetic tape. Contains abstracts of articles appearing in over 500 journals, newsletters, tabloids, proceedings, and transactions covering computers, telecommunications, and electronics.

Computer literature index. (1971). Phoenix, AZ: Applied Computer Research, Inc.
Quarterly provides subject access to practical and user-oriented computer literature appearing in eighty periodicals, books, and reports.

Current index to journals in education. CIJE. Phoenix, AZ: Oryx Press.
Available online and in CD-ROM, as part of the ERIC database. Probably the best single source for finding references to evaluations of educational software at any level.

Microcomputer index. (1980-). Mountain View, CA: Database Services, Inc.
Bi-monthly. Also available through Dialog Information Retrieval Services, File 233. The index is a subject and abstract guide to articles from fifty microcomputer journals. Publications indexed include *Byte, InfoWorld, Personal Computing,* and *MacWorld.*

REFERENCES

Askey, J. (1987). Criteria for selecting software. In J. E. Herring (Ed.), *The microcomputer, the school librarian and the teacher* (pp. 101-115). London: Clive Bingley.

Beaubien, D. M.; Emerton, B.; Kesse, E.; Primack, A. L.; & Seale, C. (1988). Patron-use software in academic library collections. *College & Research Libraries News, 49*(10), 661-667.

Campbell, P. B. (1987). Preliminary guidelines for selecting computer software. In S. Berman (Ed.), *Alternative library literature, 1984* (pp. 164-165). Jefferson, NC: McFarland and Company.

Cargill, J. (1987). Paying attention to basics: Selecting microcomputer hardware and software. *Technicalities, 7*(11).

Chiang, K. (1987). Collection policy. In H. Curtis (Ed.), *Public access microcomputers in academic libraries: The Mann Library model at Cornell University* (pp. 35-48). Chicago, IL: American Library Association.

Chiang, K., & Curtis, H. (1987). The selection and acquisition of software and computerized data. In H. Curtis (Ed.), *Public access microcomputers in academic libraries: The Mann Library model at Cornell University* (pp. 49-58). Chicago, IL: American Library Association.

Clyde, L. A., & Joyce, D. J. (1985). Selecting computer software for school libraries. *School Library Media Quarterly, 13*(2), 129-137.

Dewey, P. R. (1984). *Public access microcomputers: A handbook for librarians.* White Plains, NY: Knowledge Industry Publications.

Dewey, P. R. (1988). Looking at review sources. In S. S. Intner & J. A. Hannigan (Eds.), *The library microcomputer environment: Management issues* (pp. 63-76). Phoenix, AZ: Oryx Press.

Dewey, P. R. (1990). *Public access microcomputers: A handbook for librarians.* Boston, MA: G. K. Hall.

Dudley, C. C. (1986). Microcomputer software collection development. *Choice, 23*(5), 704-705.

Erhmann, S. (1987). Unpublished paper presented at EDUCOM Annual Conference. Pittsburgh, PA.

Green, K. C. (1990). *Results of the USC/EDUCOM survey of desktop computing.* Unpublished paper presented at the EDUCOM Annual Conference, Atlanta, GA.

Hannigan, J. A. (1985). The evaluation of microcomputer software. *Library Trends, 33*(3), 327-348.

Intner, S. S. (1988). Developing software collections. In S. S. Intner & J. A. Hannigan (Eds.), *The library microcomputer environment: Management issues* (pp. 3-21). Phoenix, AZ: Oryx Press.

Johnson, M. (1990). Adding computer files to the research library: Issues in collection management and development. In C. Gould (Ed.), *Computer files in the research library* (pp. 3-13). Mountain View, CA: The Research Libraries Group.

Jones, N. B., & Vaughan, L. (Eds.). (1983). *Evaluation of educational software: A guide to guides.* Chelmsford, MA: Northeast Regional Exchange.

Nollan, R. (1986). *Microcomputer software policies in ARL libraries* (SPEC Kit No. 123). Washington, DC: Association of Research Libraries.

Polly, J. A. (1985). Selecting really excellent software for young adults. *Voice of Youth Advocates, 8*(2), 114, 158.

Rappaport, S. (1985). Software collecting: Method for madness. *Library Journal, 110*(6), 56-57.

Schueckler, L. M., & Shuell, T. J. (1989). A comparison of software evaluation forms and reviews. *Journal of Educational Computing Research, 5*(1), 17-33.

Seiden, P. (1987). The elusive mountain: Software resources for higher education. *Academic Computing, 1*(Spring), 10-13, 61-63.

Seiden, P. (Ed.). (1988). *Directory of software sources for higher education: A resource for instructional applications.* Princeton, NJ: Peterson's Guides.

Seiden, P. (Ed.). (forthcoming). Survey of libraries providing locally mounted databases. *RASD Occasional Paper.*

Sissine, F. (1988). Using hard copy to find software: A guide to printed resources

for discovering and evaluating PC software. *Library Software Review, 7*(4), 234-240.

Swigger, K. (1986). Relating software to instruction: Problems and resources in software selection and evaluation. In B. H. White (Ed.), *Collection management for school library media centers* (vol. 7), (pp. 281-294). New York: The Haworth Press.

Troutner, J. (1989). Developing a curriculum oriented software collection. *Emergency Librarian, 16*(3), 15-16.