

ONLINE DATABASES

In-House Databases II: Evaluating & Choosing Software

By Carol Tenopir

LAST MONTH (*LJ*, April 1, p. 640-41) the column discussed sources for finding out about what software is available for the creation of in-house databases. It surveyed the growing number of software directories aimed specifically at library or information applications. These directories make the software identification process easier. Most of them, however, provide only a limited amount of evaluative information and the information about each package is only as good as the information-gathering technique used by the compilers.

Reviews of software packages are also getting easier to find, aiding the evaluation process. A forthcoming publication from Nolan Information Management Services (21203 Hawthorne Boulevard, Suite 5323, Torrance, California 90509) will bring together reviews of many information applications software packages for microcomputers. These reviews have been solicited from current users of the programs and will include samples of screen displays and printouts as well as evaluative comments. This kind of tool, together with the reviews of library packages for all sizes of computers that are included in journals (notably the twice-yearly *Software Review*), will help greatly in the preliminary evaluation of software.

This column will concentrate on evaluation of software for in-house databases. It is a complex and often time-consuming task, but a necessary one to ensure purchase of software that will do what you need and want to do.

Needs assessment

Most experts agree that evaluation of specific software packages for your in-house database should begin after you have completed a thorough needs analysis. No one or two software packages are best for everyone, because

"best" is very much dependent on your individual needs, priorities, and constraints. Although the topic of needs analysis cannot be completely covered in this column, I will outline some of the major considerations that should be included in your needs assessment and preliminary written specifications.

More information on needs assessment and specifications for the creation of online databases in the library will be found in a forthcoming publication from the Special Libraries Association. This book on how to build a private database will be available later this year and will cover all aspects of in-house database creation for libraries or information centers. For more information contact: Marjorie Hlava, Access Innovations, Inc., P.O. Box 40130, Albuquerque, New Mexico 87196 (505) 265-3591, or the Special Libraries Association. Many books on automation in the library also include discussions of needs analysis that are relevant to planning in-house databases.¹

The first question to address in the needs analysis is: "Why do we need an online database?" Stating the reasons will help define the purpose and potential users of the database, as well as make you examine the decision to automate. Some reasons for creating an in-house database system may include: increased access to library materials, the ability to offer special services that are too difficult with a manual system, facilitated information sharing among branches or among libraries, increased bibliographic control over different types of materials, the elimination of bulky and hard-to-maintain card files, speedier access to and updates of heavily used information, or consolidation of many separate referral files.

If the database is replacing an existing manual system, you should describe the existing system and how the automated version will replace or interact with it. It is also important to consider the opposite—how this in-house database will interact with other library functions. It could be an independent database of referral information that will never interface with anything else, but more likely the creation of an in-house database will allow you to offer expanded services and eventually build other databases.

Once this preliminary definition of purpose and role is done, you will want

to detail what the database will look like and what the system should do. The major considerations in this phase are:

1. The size of the database (number of records times number of characters per record).

2. The fields in each record (number of fields, what kind of information goes in each field, maximum size needed for each field, how often each field might need to be repeated).

3. The anticipated users of the database and where they will be using it. (If only librarians will access the system, it can incorporate more sophisticated search procedures than if the public is using it. Location of terminals will have hardware implications.)

4. The functions the system must perform (this should be detailed and specify how quickly the system should perform each function).

5. The priorities of those functions. (Decide what are absolutely required features, what could be implemented later, and what would be nice but are not required. Rank the features within the last two categories. At this stage don't leave out what you really need because you think it might not be available, but don't add features to the required or high priority list that are not really necessary to achieve the goals of the system.)

6. List any special requirements that must be met (OCLC tape input, ALA character set, etc.).

7. Project how fast the database will grow (anticipated number of items to be added each year minus the number of items to be eliminated).

Constraints

The initial needs assessment should include an analysis of your situation and any constraints with which it is burdened. These might be constraints of budget (what is a reasonable price range for the purchase of software), or of personnel (how many people at what level are available in the library to work on this project).

Any hardware requirements or constraints must be identified before you examine software. In the ideal situation, you would choose the software package that best meets your database requirements and then get the computer hardware to run the software. Unfor-

Carol Tenopir, formerly Library Systems Librarian at the University of Hawaii at Manoa and a former Vice President of Operations at Cibarelli & Associates, is currently a doctoral student at the Graduate School of Library and Information Science at the University of Illinois. She is a research assistant at the Information Retrieval Research Laboratory at the University of Illinois College of Engineering

tunately, in most cases this is not possible because there is an existing computer in the library, city, or organization that must be used. Sometimes budget or space constraints limit the choice of computer hardware.

If there is hardware you must use, it must be described in the preliminary specifications. The computer poses many initial limitations on what software can be used. What type of computer, its capacity, operating systems, programming languages supported, and amount it is to be used for other purposes, will all impact your software choices. Software vendors should be able to look at the hardware description and tell you whether or not their package will run on your computer.

If there is no existing computer, you should examine space available in the library for new hardware. Larger computers require environmental controls, special rooms, and lots of space. Microcomputers can be put almost anywhere in the library and will take much less space.

The software evaluation

Once you know the constraints under which you must operate, what you need in an in-house database system, and what the database should look like, you can begin to evaluate the available software packages. There are a number of useful articles on evaluation of database software² as well as workshops sponsored by Special Libraries Association, American Society for Information Science, Battelle Laboratories, and others.

Evaluation is a multiple step process. Software packages that have been identified as possible interest are evaluated using your preliminary specifications, eliminating those that don't measure up. Then the specifications are revised, and a final evaluation of the remaining software packages will narrow the choice to the one package best for you at the most reasonable cost.

Evaluation at each stage should be done by matching the capabilities and characteristics of each software package against your requirements. Packages should be compared with each other only after each package has been evaluated in terms of how it meets your needs. Some kind of checklist or evaluation form is recommended as a guide to ensure consistent and comparable evaluation of each package. This evaluation form can be as simple as an entry for each characteristic listed in your written specifications, followed by an assigned weight based on your priorities. As you examine each feature of each software package, you assign a numeric score reflecting how each feature meets your needs. The sum of each weight times each score gives you a total score for each package. (This score

in many cases relies on your subjective judgment, so it should be used as an approximate guide only.) Matthews discusses assigning dollar values to each feature and incorporating total system cost comparisons into the evaluation.³

Evaluation of software for in-house databases is tied to each library's needs assessment, but there are also basic criteria applicable in every evaluation. These criteria include: vendor or producer, software constraints and flexibility, query language, security, output capabilities, input capabilities and procedures, documentation, training, hardware constraints, and costs.

The vendors or producers of a software package can have direct and indirect impact on the success of your database system. Their awareness of a library's special database needs, their ability to talk with you in your terms, their responsiveness to queries, and their reputation and stability in the marketplace all are important factors to be included in your evaluation. Much of this information is subjective and is gathered by talking to colleagues or by analyzing your reactions to the firm.

Other factors to be evaluated under vendor include:

—Does the vendor actively market to libraries so they understand the library needs, or are they a general software house?

—How many libraries have purchased the database software package; will the vendor provide names so you can talk with users; are they using the software for a database similar to yours?

—How long has the vendor been in business; do they have a history of keeping products on the market and supporting them?

—Is there a user group or a vendor-sponsored newsletter about this software?

—What kind of support services are offered (eg. installation on your computer, modifications, conversion) and at what cost? If the vendor is a library that is selling its database software, they may not be able to provide any support services.

Software and system constraints and lack of flexibility may eliminate some software packages at the early evaluation phase. Things that should be evaluated under this heading include:

—Can this package be run on more than one kind of computer? (Some vendors offer several versions of their software; some packages will run on any computer that uses a certain operating system such as "CP/M," now standard for many microcomputers.)

—Are there upper limits to the number of records in the database, or number of fields in a record, or number of characters in each field?

—Does the vendor continually im-

prove or refine the software, and will you get the updated versions?

—Is the software in modules so changes can be made without adversely affecting other parts of the program or so you can purchase only those modules of interest to you?

—Is the program written in a standard programming language? (This is important for making modifications.)

—Will the vendor make modifications if you need them; can you make modifications; is the "source code" (necessary for modifications) available; will changes affect your warranty? (Warning: Even if the vendor says you can make modifications and will sell the source code, it is often difficult and ill-advised to modify a complicated software package.)

Evaluation of software constraints and flexibility is something you might feel uncomfortable doing without help from someone with a data processing background, but evaluation of a *query language* or searching procedures is something most librarians will feel comfortable doing.

How a user is to request information from a system is of vital importance to the success of that system. Success in design of the user-system interaction depends very much on how familiar your users are with automated systems, how often they will access this system, their ages, and their information background. More and more literature discusses "user-friendliness" in database systems. Some things to look for in a software package are:

—Is the query language easy to learn and appropriate for your users?

—Are there several levels of commands possible for different levels of users?

—If improper search language or commands are used, are there intelligible error messages that provide possible alternative actions?

—Can terms be easily combined with Boolean operators (and, or, not); with relational operators (less than, greater than, equal to); with word proximity?

—Can most users perform a single search (such as finding an article by an author) without lessons first?

Security, output, & input

Security needs will vary with each situation. Every system, however, should have password safeguards to ensure the database cannot be altered by unauthorized people. Different levels of passwords, plus a browse-only mode requiring no password, are ideal. Some libraries may want to be able to monitor use by password or by terminal for both security and management reasons.

Output is what users see when they have entered a successful search. It should be evaluated in terms of readabi-

lity of format, flexibility of format (e.g., the ability to specify that only parts of a record be displayed), and special capabilities. Special capabilities include the ability to generate printed lists or reports as needed (e.g., acquisitions lists, partial holdings lists), and types of management or statistical information on the size and use of the database.

Input capabilities are often part of the system design. Many packages allow you to determine what data input will be like based on the field characteristics defined in your needs analysis. Flexibility and capabilities will vary, however, and should be considered in the evaluation. Some things to look for regarding input are:

- Are a variety of input means possible? (You might have some machine-readable records on magnetic tape or disk that will form part of your database, and you may want to enter new information directly at the keyboard.)

- Are the procedures for adding new items easy to learn and are there prompts so the person doing the input knows what to enter next?

- If an input error is made, will the system alert the person doing the input?

- Can entire records or partial records be added, changed, or deleted?

- Can different input procedures and structures be designed for different types of materials?

Documentation & training

Many of the principles of software documentation evaluation are well known to librarians because they are the same as the evaluation of other reference materials. Software documentation should be written at an appropriate level of sophistication for the intended users; be of sufficient length to describe the software in detail; include appropriate illustrations and examples; be organized in a logical way; and include an index, glossary, and table of contents. Revisions or updated versions should be provided. Large systems should come with several levels of documentation—for end users, librarians, and data processing personnel. Microcomputer packages should also have clear and readable documentation. Before you do the final software evaluation, you might want to purchase the documentation for the few most likely software packages. This will answer many of your questions about the software as well as allow evaluation of the documentation.

Training is related to the documentation, but it also might include online lessons, training sessions at your library, or classes held at the vendor's facility. The cost for the amount of training needed for your staff and your users should be estimated because it could add greatly to the total system cost.

While evaluation of computer hardware is outside the scope of this discussion, it will also be a factor in your in-house database, especially if you are purchasing a microcomputer. As mentioned, ideally you should first choose the software that meets your needs, then purchase hardware that will run it. (There will still be many choices, however, and you may want to seek help at this stage.)⁴ Some evaluation criteria related to hardware include:

- Are special terminals required, and if so, are they readily available and cost competitive?

- How many users can the system support at one time? Can this be expanded? Is dial-up access available?

- How will peak use affect response time? What are average and peak use response times for databases comparable to yours?

Finally, software choice for every application comes down to cost. Total costs based on the level of effort in implementing a database on a given system should be considered in addition to out-of-pocket costs. However, these may be difficult to estimate. The software lease or purchase price, plus maintenance costs, training, additional services, conversion, equipment and any costs for software updates should at least be compared. If costs for level-of-effort can be assigned, these will be added to the above costs to calculate total system costs. Finding the software package that rates highest on the evaluation for the lowest cost is the goal of the evaluation process.

Evaluation of software for an in-house database is obviously not a simple matter. It requires time to find out about different packages and careful consideration of many factors. If it is done carefully, however, implementing your database will go much more smoothly.

Software packages

The directories of software for library applications list dozens of commercially available software packages for in-house databases, so every package for every size database cannot be listed here. Because microcomputer-based systems are the newest systems and they allow even small libraries to create in-house databases, several microcomputer software packages for the creation of in-house databases are listed below:

Golden Retriever

Golden Retriever operates on a TRS-80 Model II with three floppy disk drives or a Winchester hard disk and 64K memory. This package has been widely reviewed in the literature and is in use in many libraries. A demonstration disk can be obtained from CLASS

for \$50. Software costs for a database of up to 4500 records are under \$2000.

Available from: CLASS, 1415 Koll Circle, Suite 101, San Jose, CA 95112 (408) 289-1756; Capital Systems Group, Inc., 11301 Rockville Ave., Kensington, MD 20795 (301) 881-9400; or Gaylord, P.O. Box 4901, Syracuse, NY 13221 (800) 448-6160.

Star

Star operates on an Alpha Micro with a hard disk. It has been described in the library literature and is a powerful and highly regarded software package. It incorporates sophisticated searching features of the large commercial systems and can accommodate large or small databases. Software price is approximately \$20,000.

Available from: Cuadra Associates, Inc., 2001 Wilshire Blvd., Suite 305, Santa Monica, CA 90403 (213) 829-9972.

BRS/Search for Micros

BRS/Search for Micros was introduced late last year, this package is a micro/mini version of the powerful BRS/Search software. It runs under the UNIX operating system and currently runs on the Onyx and WICAT microcomputers as well as PDP/11 and VAX minicomputers. Software costs vary with the hardware configuration but are in the \$5000-\$30,000 range.

Available from: BRS Software Group, 1200 Route 7, Latham, NY 12110 (800) 833-4707.

Inmagic

Inmagic is a new CP/M micro version of the popular minicomputer software. Version 1 of Inmagic for micros was first available in April of this year. Version 2 should be ready this summer to run on Televideo, Superbrain, and DEC Rainbow micros; a version for the IBM Personal Computer will be available later in 1983. Cost for Version 1 is under \$1000, plus \$250 for a separate Report Generator.

Available from: Warner-Eddison Associates, Inc., 186 Alewife Brook Parkway, Cambridge, MA 02138 (617) 661-8124.

Other microcomputer software (mostly in the \$2000-\$3000 range) for in-house databases include:

CARD DATALOG

Card Datalog is for any microcomputer with a CP/M operating system. It is based on the dBase II Data Base Management System (DBMS). Also includes acquisitions, circulation, and series modules.

Available from: DTI Data Trek, 121 W. E Street, 2nd floor, Encinitas, CA 92024 (714) 436-5055.

Computer Cat

For Apple II Plus or Bell and Howell computers.

Available from: Computer Cat, 3005 W. 74th Avenue, Westminster, CO 80030 (303) 426-5880.

Gaylord System

For Apple II Plus computers; includes acquisitions, circulation, film booking and the CLASS Checkmate serials system modules.

Available from: Gaylord, P.O. Box 4901, Syracuse, NY 13221 (800) 448-6160.

References

1. See for example: Joseph R. Matthews, *Choosing an Automated Library System: A Planning Guide*, ALA, 1980; John Corbin, *Developing Computer-Based Library Systems*, Oryx Pr., 1981; and Richard W. Boss, *The Library Manager's Guide to Automation*, Knowledge Industry Publications, 1979.
2. Garoogian, Rhoda. "Prewritten Software: Identification, Evaluation, and Selection." *Software Review*, February 1982, p. 11-34; Carol L. Norris and Diana Marincola. "Guidelines for Developing an Online In-House Database Through a Commercial Vendor." *Proceedings of the Third National Online Meeting* (1982): p. 435-444; "An Overview of the Make vs. Buy Decision." In: *Applications Software Solutions*, Delran, NJ: Datapro Research Corp., 1979.
3. Matthews, Joseph R. *Choosing an Automated Library System*. ALA, 1980.
4. One source of help is the *Terminal/Microcomputer Guide and Directory*, 3d ed. Weston, Connecticut: Online, Inc. Annual supplements.

SYSTEMS, AUTOMATION, & LIBRARIES

BUY, BUILD, ADAPT—OR FORGET IT!

By Susan Baerg Epstein

A LIBRARY considering an automated circulation control system or an online catalog has a large number of choices as to the hardware and software to be used to perform these operations. Basically, these options can be considered in one of three categories: 1) purchasing a turnkey system; 2) adapting some other library's system to run on the purchasing library's hardware; or 3) building the library's own custom system.

Buying

A "turnkey" system is one in which a single source provides the computer (hardware), computer programs (software), and maintenance (of software and/or hardware). Theoretically, these systems are delivered completely operational, ready to be plugged in and turned on; hence the term "turnkey." In reality, the process is not that simple or painless, but it is probably less painful than any of the alternatives. Turnkey systems can be single purpose, such as a circulation system, or "integrated," such as an acquisitions subsystem, a circulation subsystem, and an online catalog subsystem using the same computer and the same database. A turnkey system is defined not by the functions it performs, but by the single source for all system components. Current circulation system vendors include CL Systems, Inc. (CSLI), Computer Translations, Inc. (CTI), DataPhase Systems, Inc. (DSI), Data Research

Associates, Inc. (DRA), Gaylord, GEAC, and Universal Library Systems (ULISYS).

The major advantage of a turnkey system results from the distribution of software development and maintenance costs among many users. In order for this to occur, it is necessary for the vendor to supply virtually the same system to each user. Needs of individual libraries are handled not by writing custom computer programs for each library, but by writing a general program which obtains its specific values, such as length of loan period for a circulating book, from a table. The information in the table is provided by the individual library. When the program is operating, it looks to this table of parameters for the necessary information, much like "filling in the blanks" for the computer program. These parameter tables also mean that a library can easily change the options, such as changing the standard loan period from 14 to 21 days or changing a fine from 5¢ to 10¢ a day, without actually changing the computer program itself. Thus, library personnel can make the changes without vendor or programmer intervention. Each of the turnkey circulation vendors differs in the degree to which parameters are used and in the ease of change.

Some libraries have decided not to use a turnkey vendor because they feel that their needs are so unique that a circulation system which can accommodate the needs of any other library cannot possibly accommodate their needs. This is rarely the case. There is a tendency among these libraries to attempt to replicate their manual procedures on automated systems, step by step. If a library attempts to do this, it will not only be unsuccessful in the replication, but will not utilize the auto-

mated circulation system most effectively. Manual circulation systems have limited procedural options due to staff cost or time. An automated circulation system presents a library with a multitude of options due to the computer's speed and ability to maintain large files and only notify library staff of the exceptions. (For instance, a circulation file is kept only to be able to locate a specific book and to determine which materials are overdue, not for the 98 percent of the transactions which are never consulted for either purpose.) A library should take advantage of the automated system's potential by examining *why* a certain procedure or policy is used, not *how*. Once the *why* is determined (and this is sometimes difficult since no one on the library staff will remember why a particular procedure was instituted and the library will be reluctant to change something that has "always been done that way"), then the library can determine *how* the automated circulation system can be used. A library which wishes to make the most effective use of its automated circulation system should examine every policy and procedure and expect to make major changes when the automated circulation system is implemented.

Sometimes a turnkey vendor will agree to install a custom feature for an individual library. In most cases, the vendor will then incorporate this new feature in the vendor's general turnkey system and offer the feature to other libraries through the vendor's software maintenance program. If this occurs, the library can be certain that this original custom feature will be maintained and corrected at the same level as the vendor's other software. However, this sometimes does not occur. The library's best protection is to specifically

Susan Baerg Epstein is president of Susan Baerg Epstein, Limited, a consulting firm that specializes in library automation and management located in Costa Mesa, California