

# Evaluation of Healthcare Software from a Usability Perspective

**Julie C. Lowery and James B. Martin**

*This paper provides a framework for evaluating healthcare software from a usability perspective. The framework is based on a review of both the healthcare software literature and the general literature on software usability and evaluation. The need for such a framework arises from the proliferation of software packages in the healthcare field, and from an historical focus on the technical and functional aspects, rather than on the usability, of these packages. Healthcare managers are generally unfamiliar with usability concepts, even though usability differences among software can play a significant role in the acceptance and effectiveness of systems. Six major areas of usability are described, and specific criteria which can be used in the software evaluation process are also presented.*

## INTRODUCTION

This paper addresses the evaluation of healthcare software from a usability perspective. Shackel<sup>1</sup> defines usability as “the capability to be used by humans easily and effectively.” This concept differs from functionality, which is “the capability in machine functional terms to fulfill the specified range of tasks within the specified range of environmental scenarios.” While a software package might be quite functional, in that it can perform desired tasks, the *manner* in which it performs these tasks may be cumbersome and confusing—that is, the system may not be usable. As software packages in the healthcare field have proliferated, usability issues have become critical in the evaluation of alternative systems.

Healthcare facilities are purchasing more and more predesigned or packaged software systems.<sup>2,3</sup> At the same time, the number of vendors from which systems can be purchased is also increasing. While much has been written on the design and selection of healthcare software,<sup>4-14</sup> the guidelines and recommendations previously provided are too general to be of practical use. In addition, the focus of many existing guidelines is on the technical and functional aspects of systems, rather than on system usability. As a result,

---

*From the Health Services Research and Development Field Program, Veterans Administration Medical Center, Ann Arbor, Michigan 48106; and the Department of Health Services Management and Policy, School of Public Health, The University of Michigan, Ann Arbor, Michigan 48109.*

healthcare managers are generally unfamiliar with usability concepts and may have difficulty identifying usability differences between systems, even when these differences are significant.<sup>15</sup> Because a system's usability can have a significant impact on overall system effectiveness, it is important that purchasers of healthcare information systems understand usability concepts, as well as understand how these concepts should actually be integrated into software packages.

This paper provides a framework for evaluating alternative healthcare software systems from a usability perspective. Included for each category of this framework are specific criteria which can be used in the selection process. Some of these criteria should be included in a Request for Proposal (RFP), while others are more easily addressed during an actual system demonstration. The application of the criteria presented in this paper should result in the selection of systems which enable users to perform required functions more easily and effectively.

## DESCRIPTION OF USABILITY FRAMEWORK

The usability issues presented in this paper were gathered from the healthcare software literature as well as the general literature on software usability and evaluation. While the issues presented here are not new, the framework used to categorize these issues is new, and is believed to offer an improved manner in which to conceptualize usability.

To date, usability has not been addressed comprehensively in the healthcare software literature. The general software usability literature has, however, categorized the major usability issues in a variety of ways,<sup>16-18</sup> as depicted in Table 1. In addition to the three summarizations shown in Table 1, various authors have written on particular usability

**Table 1. Categorization of Usability Issues**

Authors (Reference)	Categories
Rubinstein and Hersh <sup>16</sup>	<ol style="list-style-type: none"> <li>1. Conceptual model</li> <li>2. Language</li> <li>3. Learning aids</li> <li>4. Human interface styles</li> <li>5. Responding to users</li> <li>6. Presentation and representation</li> </ol>
Whiteside, Jones, Levy, and Wixon <sup>17</sup>	<ol style="list-style-type: none"> <li>1. System feedback</li> <li>2. Consistency of input forms</li> <li>3. Usefulness of help systems</li> <li>4. Naturalness of input forms</li> <li>5. Navigation through the system</li> </ol>
Petreley <sup>18</sup>	<ol style="list-style-type: none"> <li>1. Performance and features</li> <li>2. Documentation</li> <li>3. Ease of learning</li> <li>4. Ease of use</li> <li>5. Error handling</li> </ol>

issues, including system dialogue,<sup>19</sup> query languages,<sup>20-24</sup> human interface styles,<sup>17,25</sup> system message design,<sup>26</sup> and information display.<sup>27</sup>

The usability framework presented in this paper includes all of the issues just mentioned, but recategorizes them in the following manner: (1) Logical Organization of Procedures, (2) Screen Design for Data Entry, (3) Error Handling, (4) Data Retrieval/Report Generation, (5) Learning/Help, (6) Consistency. The objectives in developing this framework were (1) to stress usability issues of particular importance in the system evaluation and selection process (e.g., error handling and data retrieval/report generation); and (2) to consolidate usability issues into categories more meaningful to healthcare managers.

The remaining sections of this paper describe each of the six categorization areas, and include a discussion of the treatment of each area in both the healthcare software and general software literature. In several instances, important usability issues are addressed minimally in both sets of literature, and these issues are hence addressed in more depth herein. In addition, specific evaluation criteria are presented in each section. It is recommended that these criteria be used by individuals responsible for evaluating alternative systems; the term "user" is employed (rather than "evaluator") in referring to these individuals throughout this paper, in order to emphasize the importance of evaluating a system from the perspective of one who will actually be using it.

## LOGICAL ORGANIZATION OF PROCEDURES

The procedures for performing a task within a system should follow a logical organization that is meaningful to the user. A logical organization helps the user understand and predict the system's behavior, which not only reduces the time required to learn to use the system, but also increases the user's satisfaction with the system.

The healthcare software literature fails to discuss logical organization as a usability issue. In the general literature, Whiteside *et al.*<sup>17</sup> found that many of the problems users had with experimental systems were the result of not understanding the logic by which the system performed various tasks. The results of a study conducted by Badre<sup>27</sup> emphasized the importance of presenting information and tasks in a manner which is meaningful to the user. Consequently, he recommends that the user's information processing habits be studied and understood in order to design (or select) a usable system.

**Evaluation Criteria.** Several criteria should be used for evaluating a software system in terms of the logical organization of procedures. Frequently, an actual demonstration of the various system tasks is required for a satisfactory evaluation to be completed. One way to determine whether the manner in which a system performs a given function is meaningful to the user(s) is to compare the system's procedures with the current, manual procedures used for performing the function (in cases where the software system under evaluation is to replace a manual function). Because users are likely to be familiar and comfortable with the current, manual procedures, the extent to which the software system mimics these procedures will naturally facilitate the learning and acceptance of the new system.

It should be cautioned, however, that comparison of current procedures with a proposed system's procedures is only useful if it has been determined that the current

procedures are the best method of performing the task in question. The procedures comprising each required task should be reviewed prior to automation to determine whether or not they should be altered to improve their efficiency and/or effectiveness. If such a review results in a decision to change current procedures, then the new procedures should, of course, be used as the basis for comparison against the manner or sequence in which tasks are performed by the proposed software.

The logic of a system's overall organization should also be reviewed. Vendors should be asked to provide a schematic representation, much like an organizational chart, of the relationship between the different menus within the system. The relationship between menus, and between submenus within each menu, should make sense to the user. In addition, the same functions should not be listed in more than one menu within a module, unless such redundancy is logically appropriate.

A third criterion which can be used in evaluating a system's organization is the manner in which the system helps the user understand the organization/sequence of procedures. A system should, for example, provide information to the user on procedures just performed, to help determine the procedures that should be performed next. For example, menu-driven systems usually require the user to step through several different menus in order to accomplish a particular task. The Whiteside *et al.* study<sup>17</sup> found that "in the menu interface, many users became so involved with maneuvering through the menu structure that by the time they had done this successfully, they forgot what the task was." Some systems provide "windowing" menus, which layer on top of each other as one progresses further into the menu structure. This feature makes it easy for the user to understand exactly where (s)he is in the task.<sup>28</sup>

## SCREEN DESIGN FOR DATA ENTRY

The presentation of information on the screen of the user's video display terminal (VDT) plays a significant role in the usability of a system. While some of the literature on healthcare software design and selection mentions the importance of screen displays for data entry,<sup>8,9</sup> rarely is any further explanation provided as to what constitutes a good screen display. In the general software literature, the importance of the spacing and sectioning of items on a screen is discussed. Usually the liberal use of spacing and sectioning is preferred to a cluttered screen containing too much information.<sup>16</sup> It is also recommended that data entry screens support the user's initial or habitual response tendencies (i.e., screens should appear "natural" to the user).<sup>17</sup> Screen formats which closely resemble the hardcopy documents originally used to capture data (prior to implementation of the computer-based system), or which resemble the hardcopy documents from which data are being transcribed to the system, generally meet this criterion of "naturalness" of input screens.

**Evaluation Criteria.** In selecting a packaged software system, users may find that they have little say in designing the data entry screens to closely resemble their organization's documents. In such cases, users should look for data entry screens that are easy to read and that follow as closely as possible the organization's currently established sequence for recording/entering data. However, as healthcare software becomes increasingly sophisticated, users should look for "screen painting" features, where the user can

design his/her own screen. A number of personal computer database packages available in the general market already have such features.<sup>18,29</sup>

## ERROR HANDLING

An important component to an effective information system is the detection of errors—in the procedures required to perform a task, as well as in the data entered into the system. The detection of errors is especially critical if the data are to be used and accepted as valid throughout an organization. The following discussion of error handling features in software systems is divided into three sections: (1) detection and correction of procedural errors; (2) detection and correction of errors during data entry; and (3) detection and correction of errors following data entry.

### Procedural Errors

The healthcare software literature pays scant attention to the issue of procedural errors, while the general software literature gives more consideration to the subject. If a user makes a procedural error (e.g., enters the wrong syntax for a command or presses the wrong key), the error message provided by the system should provide sufficient information so that the user can understand the nature of the error and how to correct it. Irrelevant messages related to the internal workings of the software should not be displayed. If possible, examples of how to correct the problem should be provided.<sup>17,24</sup> Some systems may even attempt to correct the error.<sup>16,18</sup>

**Evaluation Criteria.** During vendor demonstrations or site visits, procedural errors should be demonstrated, so that the system's responses can be evaluated with respect to the standards described in the preceding paragraph.

### Errors During Data Entry

Unlike many other usability issues, the issue of data edits, both during and after data entry, has been addressed in some detail in the healthcare software literature—perhaps because of the critical need for accurate data. Waters and Murphy<sup>9</sup> present seven different data validation tests that can be conducted by systems to ensure the accuracy of the data. Martin *et al.*<sup>30</sup> discuss specific data edits and cross-checks which are performed in a functioning Surgical Suite Management Information System.

In evaluating the data entry features of general database software, the general computing literature looks carefully at the power of the data error detection facilities. Systems that allow the user to specify a wide variety of editing criteria, including the type of data required by a field (e.g., text, numbers, dates, money, etc.), the format of the data, and valid data values, are ranked more highly than those systems which allow the user to specify more limited criteria.<sup>16,18</sup>

**Evaluation Criteria.** Vendors should provide a detailed list of the criteria against which the data for each field are checked. Clearly, the more fields on the list and the more specific the criteria, the better. In addition to these vendor-supplied edits, the users should be able to build—or modify over time—their own facility-specific edits.

When an error is made in data entry, the system should notify the user of the error

in a clear, yet inoffensive, manner. A short beep or flashing of the field in question is effective. Once the system has notified the user of the error, it should provide an explanation of the nature of the error and of the possible means for correcting it. A list or description of valid entries is a possible effective response.

If an error is detected, but the user is unable to identify the correct data, the system should allow the user to override the error correction facility and continue with data entry, rather than prevent the user from continuing. The system should subsequently produce a report which identifies those fields with missing or erroneous data (see "Evaluation Criteria" for next section).

### **Errors Following Data Entry**

Certain data errors cannot be detected during data entry, but can be detected at a later time. Errors of this type are detected by comparing data between fields, records, or files/systems. For example, information systems which collect sequential time data on a series of tasks or activities can compare the data between fields, to ensure that the times are sequential. An example of a cross-check between records is to verify that only one record exists for a particular patient (when only one is allowed). Cross-checks between files/systems are also important for ensuring data validity and completeness. For example, for every case admitted through a hospital admissions system, a record should exist within the hospital billing system. A comprehensive information system consisting of multiple files or systems should include a comprehensive set of such cross-checks.

**Evaluation Criteria.** Vendors should be asked for a listing of all those fields which are cross-checked following the entry of all fields in a record, as well as following entry of all associated records. Included in the vendor list should be a description of the type of cross-check performed—against other fields, records, or files/systems. In general, the greater the number of cross-checks, the more valid the data will be. As with edits during data entry, the ability of the users to define and update their own cross-edits is a highly desirable feature.

The reports which present the results of these cross-checks (sometimes referred to as "data control reports") are key to the ease with which corrections can be made. Since both the healthcare and general software literature provide little discussion of such reports, appropriate content is overviewed here.

Data control reports should present detailed information regarding: (1) any errors (including missing data) not corrected during the data entry edit-checking process; and (2) errors detected through the process of cross-checking between fields, records, or systems. These reports should include (1) any data necessary for locating associated hardcopy records; (2) record keys (so that the record can be retrieved and corrected); (3) designation of the field with the erroneous or missing data; (4) the erroneous data values; and (5) space for recording the correct data (to facilitate subsequent correction).

## **DATA RETRIEVAL AND REPORT GENERATION**

An area of particular interest to hospital administrators is reporting. As information systems and their associated databases proliferate, the amount of data available to man-

agement for decision-making purposes also proliferates. The ease with which management can obtain and understand these data is thus an important usability issue.\* This issue consists of three major components: (1) standard system reports; (2) ad hoc query; and (3) ad hoc report generation. Each of these topics is discussed below.

### Standard System Reports

Standard system reports are pre-programmed by the vendor and, therefore, do not need to be produced using the ad hoc data retrieval capability (discussed in the next section). These reports are advantageous for complex calculations, or when data from more than one data base are required. The retrieval of data for such reports can be difficult, even with a powerful data retrieval capability. Therefore, systems generally include a set of standard reports for which a program has already been written to perform the necessary calculations and data base operations. The user then needs only to issue a single command to initiate the data retrieval/report generation program.

Much of the healthcare software literature mentions the importance of report content and format in the design or selection of an information system<sup>4,6,7</sup>; few guidelines are provided, however, for evaluating the reports produced by different vendors. The general software literature has focused its attention on the ad hoc query capabilities of systems, rather than on the standard report capabilities. Since this issue has been addressed inadequately in both sets of literature, it is expanded upon here.

**Evaluation Criteria.** Perhaps the easiest way to evaluate a set of standard reports is to compare them with current reports that are prepared manually, if the current reports are widely used and accepted by hospital management. To the extent that the new information system will provide data which has been previously unavailable to hospital management, it would be worthwhile for management to design several key reports which they would like to receive on a recurring basis. These designs can then be compared against the vendors' standard reports to identify those vendors who come closest to providing the desired reports.

Well-designed reports are not cluttered; presenting too much information on one report can be overwhelming and confusing, and can detract from what is important. Reports should also be clearly labeled, including the title, column and row labels, identification of which records/time period are represented, and an explanation of how calculations were performed. Managers should review all of the vendors' standard reports for clarity. If managers cannot understand a report, it is not clearly conveying the necessary information.

### Ad Hoc Data Retrieval

Ad hoc data retrieval refers to the capability to retrieve data on an "as needed" basis from the data base. Many reports cannot be pre-defined by the user and cannot, therefore,

---

\* Closely related to the issue of the retrieval and reporting of data is the comprehensiveness of the data within a system. To evaluate the extent to which a system contains the critical data required by management, an analysis of managers' information requirements should be performed. A technique that has been effectively used in the design of executive information systems is the Critical Success Factor Methodology, described by Bullen and Rockart,<sup>31,32</sup> and Thomas and Lowery.<sup>33</sup>

be incorporated into a standard reporting capability. If the necessary data are in the data base, users should be able to retrieve the data, manipulate them, use them to perform the required calculations, and then display the results in a meaningful format.

Most of the literature on healthcare software design and selection simply states that the data retrieval capability should be flexible, without offering an explanation of what constitutes flexibility. In contrast, the literature on general software usability issues has devoted considerable attention to the issue of data retrieval, particularly in the area of query languages. One of the major findings that has emerged from studies of query languages is that users should be allowed to employ their own natural strategies for querying. These strategies include specifying a series of simple queries, rather than a single, complex query. This can be effectively accomplished by selecting the various components (e.g., from a menu or in response to prompts) rather than by producing the entire query (as in writing a command).<sup>16,20-22</sup> If a command structure is required, the user should be able to use ordinary English phrases, rather than more formal logic statements.<sup>23,24</sup>

“Query-by-Example” has been found to be an effective query methodology for many users. In this methodology the user creates the general structure of the table (s)he wishes to produce, then writes the selection criteria in the proper columns (representing fields in the database) of the table. In contrast, Standard Query Language (SQL), the emerging standard language for relational database management systems,<sup>34</sup> presents difficulties for most inexperienced database users because of the formal, logically precise nature of the commands.<sup>35</sup>

**Evaluation Criteria.** In evaluating the usability of an ad hoc data retrieval capability, the vendor should be asked to demonstrate two different types of data retrieval: (1) record retrieval and (2) crosstabulation. In record retrieval, the user wishes to display or print certain fields of a subset of records in the database. The important components of record retrieval are (1) the designation of the fields that are to be displayed for each record selected, and (2) the designation of the record selection criteria.

In cross-tabulation, the system counts the number of occurrences of each unique value combination for the field categories specified. The important components of cross-tabulation are (1) the designation of the field categories that are to be used in the tabulation; and (2) the designation of the statistical computations to be performed in addition to the tabulation of the data (e.g., average, standard deviation, minimum, maximum, and total).

The results of previous studies regarding query systems can serve as guidelines for the evaluation process. Specifically, a system which lets the user specify the selection criteria by entering them in a table under the appropriate fields (as in Query-by-Example), selecting them from a menu of possible criteria, or responding to interactive prompts, is easier to use than one which requires the user to specify the query as a statement or command.<sup>†</sup> If a system does use menus or prompts to specify the selection criteria, the

---

<sup>†</sup> While menus are generally easier for less experienced users, commands are faster for more experienced users. Some software systems offer the capability to change the method of data retrieval (e.g., from menu-driven to command-based) in order to accommodate the more experienced user and increase the speed with which procedures are performed. Software with this capability is desirable if the system is to be used routinely by some people and infrequently by others.



system should display the cumulative results of each user action, in order that the user can determine whether the resultant query is the intended one.

Whatever query technique is used, it should be consistent for both types of queries (record retrieval and cross tabulation). Finally, the system should allow the user to save the resultant query in order that it can be accessed and modified, if necessary, for future use.

### **Ad Hoc Report Generation**

Once data are obtained through a system's data retrieval capability, reports presenting the data need to be generated. The report format in which data are presented plays a major role in contributing to the user's ability to understand the data. Because the usability of the resultant reports depends so much on the functionality of the system's report generation capability, the evaluation criteria for ad hoc report generation are actually functionality issues, and are discussed briefly below.

**Evaluation Criteria.** At a minimum a system should provide the capability for adding/editing titles, headers, and footnotes to the report, as well as performing other edits on the report (e.g., deleting sections, altering spacing, etc.). Users should be able to view the report, or part of it, before printing it, to ensure that the format is correct. Other desirable capabilities include specifying which fields are rows and which fields are columns; displaying subtotals for certain categories; selecting the order in which categories are presented (e.g., ascending or descending numerical, alphabetical); displaying time data in either hours or minutes; and determining the number of significant digits to be displayed.

## **LEARNING/HELP AIDS**

Learning/help aids provide instructions on how to use a system, including instructions for such activities as entering data, retrieving data, editing records, maintaining dictionaries, etc. These aids can be in hard-copy form (usually referred to as "documentation") or on-line. A discussion of these two types of aids follows.

### **On-Line Help**

The healthcare software literature has little to offer in terms of specific criteria to consider when evaluating on-line help features. The general computing software literature evaluates different systems on the extent to which a user can obtain help, and escape if necessary, at any point in the system's procedures.<sup>18,29</sup> Gaines and Shaw<sup>19</sup> recommend the capability to ask for help at two different levels. At the first level, instructional material should be available throughout all system procedures, to be accessed by the user through a simple, uniform mechanism. The material should be organized so that the user accesses brief memory aids first, but has further access to a second level of help, which provides more detailed explanations, if desired.

**Evaluation Criteria.** On-line help is especially desirable during data entry procedures, when the user may need a reminder of the possible valid entries for a field. For each

data entry field, the user should be able to press a “help” key to display an explanation and/or listing of the possible valid entries.

In addition to providing help during data entry, a system should be able to respond to users’ requests for assistance during other tasks. The method of requesting help should be consistent (e.g., use of the same function key) across tasks, and the on-line explanations should be self-explanatory. Users can evaluate the on-line help features of a system by requesting help at different points in a variety of tasks, then seeing whether the explanation provided is sufficient for completing the task at hand.

## Documentation

*The Hospital Software Book*<sup>4</sup> provides one of the more detailed lists of criteria available in the healthcare software literature for evaluating user documentation, including the following issues: organization, modularity, prioritization, accessibility, succinctness, readability, comprehensiveness, timeliness, fit and finish, relevance, orientation, and convenience. The general computing software literature pays particular attention to documentation when evaluating alternative software packages.<sup>18,28,29</sup> Desirable characteristics of documentation include a consistent tone that is neither too simple nor too complex; easy-to-read reproductions of screens and reports; high quality presentation (e.g., type set); logical organization of material (which does not require the user to refer to different sections in the text to complete one task); and the existence of both a tutorial manual and a reference manual. Gaines and Shaw<sup>19</sup> also recommend that user manuals be based on actual user/system dialogue—the use of the system should be illustrated with actual dialogue sequences that achieve specific objectives.

Rubinstein and Hersh emphasize the importance of using examples throughout the documentation.<sup>16</sup> Rather than explain in general terms how something is done, the documentation should present examples of actual menus or commands, possible user responses to the system, and the system’s responses to the user. Examples are especially important for tasks such as data retrieval, where the same data can be retrieved in a variety of different ways. The number of possible queries which users can pose is virtually unlimited, thus making it impossible to explain how to formulate every possible query. However, multiple examples of different queries will help the user understand the underlying logic of the data retrieval function.

**Evaluation Criteria.** In addition to the above recommendations, most of the usability literature recommends that documentation include the following: an extensive index (in addition to the table of contents); headers (or footers) on each page, which describe the subject matter on that page; a data dictionary for every data base into which users enter or retrieve data; a description of all possible responses, and their consequences, to each question/menu item; and a listing/explanation of all possible error messages.

## CONSISTENCY

The mechanisms for performing various tasks within a software system should be as consistent as possible across all tasks and modules. Consistency of procedures enables users to make predictions regarding system behavior, which, in turn, helps the user

understand how to accomplish specific tasks. To the extent that a system is not consistent across tasks or modules, users will make erroneous assumptions and predictions that inevitably lead to mistakes and frustration.<sup>16,19</sup>

The healthcare software literature virtually ignores the issue of consistency, while the general software literature recognizes it as an important usability issue. The general computing software literature, in evaluating various software packages, tends to group this issue with other usability issues under the general evaluation criterion of "ease of use."<sup>18,29</sup> Software packages with consistent procedures (e.g., use of function keys, movement through menus, menu layout and item selection, use of commands) are ranked higher than those lacking consistency. The Whiteside *et al.* study<sup>17</sup> of the effect of interface style on usability found that consistency in the use of function keys is positively related to usability, as is consistency in the use of syntax.

**Evaluation Criteria.** To evaluate a software package's consistency, a demonstration of the system across multiple tasks, both within and between modules, is necessary. The user should look for consistency in the use of function keys (including HELP, ESCAPE, and ENTER keys), menu item selection, stepping through menus, and command syntax. The location of key pieces of information (e.g., menu descriptions, help information) on the display screen should be the same across tasks and modules. The method for entering and retrieving data, whether it is menu-driven, command-based, or responses to prompts, should also be the same across tasks and modules.

## SUMMARY

Usability is a key concept in the evaluation of healthcare software. Non-usability issues, such as system capacity, interfacing capabilities, and vendor support services, are also important, but have not been discussed here. Most hospitals have an information systems department which can provide assistance in evaluating these areas. Usability issues, however, are often overlooked or not well understood by technically-oriented information systems staff. Managers and other users, if they do not have much experience with information systems, are also not likely to understand or identify the important usability issues without some specific guidelines. Given the importance of these issues in affecting overall user satisfaction with and acceptance of a system, and given the considerable variability in the manner in which vendors address these issues, improved guidelines for the evaluation of vendors on these issues are needed. This paper is intended to provide a set of such guidelines.

## REFERENCES

1. Shackel, B., The concept of usability, *Visual Display Terminals*, John Bennett *et al.* (eds.), Prentice Hall, Inc., Englewood Cliffs, N.J., 1984.
2. Packer, C.L., Hospitals' info spending tops \$3.5 billion. *Hospitals*, pp. 76-77, December 20, 1986.
3. Carter, K., Flexible installation, financing spur microcomputer demand. *Mod. Healthc.* 16(13):49-51, June 20, 1986.
4. Rowland, H.S., and Rowland, B.L., *Hospital Software Sourcebook*, Aspen Publishers, Inc., Rockville, Md., 1985.

5. Anderson, P., The human side of the RFP process. *Comput. Healthc.* 5(12):38-41, 44-45, December 1984.
6. Grams, M.D., Evaluation tools for hospital computer equipment and systems. *J. Med. Syst.* 8(4):229-239, August 1984.
7. Austin, C.J., *Information Systems for Hospital Administration*, Health Administration Press, Ann Arbor, Mich., 1983.
8. Lutheran Hospital Society of Southern California, *Automated Hospital Information Systems: How To Decide What You Want*, Pluribus Press, Inc., Chicago, 1983.
9. Waters, K.A., and Murphy, G.F., *Systems Analysis and Computer Applications in Health Information Management*, Aspen Systems Corporation, Rockville, Md., 1983.
10. Priest, S.L., *Managing Hospital Information Systems*, Aspen Systems Corporation, Rockville, Md., 1982.
11. American Hospital Association, *Hospital Computer Systems Planning: Preparation of Request for Proposal*, Chicago, 1980.
12. Williams, H.G., and Toole, J.E., Evaluating and selecting a computer system. *Top Health Care Finance. Health Info. Syst.* 4(4):73-91, Summer 1978.
13. Austin, C.J., Planning and Selecting an Information System, *Hospitals* 15:95-100, 202, October 16, 1977.
14. Wooldridge, S., *Software Selection*, Petrocelli Books, New York, 1973.
15. Trammel, D., There are differences in system capabilities . . . (interview), *Comput. Healthc.* 5(12):46, 49, December 1984.
16. Rubinstein, R., and Hersh, H., *The Human Factor: Designing Computer Systems for People*, Digital Press, 1984.
17. Whiteside, J., Jones, S., Levy, P.S., and Wixon, D., User performance with command, menu, and iconic interfaces. *Proceedings of the 1985 Conference on Human Factors in Computing Systems, the Association for Computing Machinery, Inc.*, New York, 1983, pp. 185-191.
18. Petreley, N., Product comparison: Relational databases. *InfoWorld* 10(16):37-52, April 18, 1988.
19. Gaines, B.R., and Shaw, M.L.G., Dialog Engineering. *Designing for Human-Computer Communication*, M.E. Sine and M.J. Coombs (eds.), Academic Press, New York, 1983, pp. 23-54.
20. Thomas, J.C., Psychological issues in the design of database query languages. *Designing for Human-Computer Communication*, M.E. Sine and M.J. Coombs (eds.), Academic Press, New York, 1983, pp. 173-208.
21. Zloof, M.M., The query-by-example concept for user-oriented business systems. *Designing for Human-Computer Communication*, M.E. Sine and M.J. Coombs (eds.), Academic Press, New York, 1983, pp. 285-310.
22. Allen, R.B., Cognitive factors in human interaction with computers. *Directions in Human/Computer Interaction*, A.N. Badre and B. Shneiderman (eds.), Ablex Publishing, Norwood, N.J., 1982, pp. 1-26.
23. Tennant, H.R., Ross, K.M., and Thompson, C.W., Usable natural language interfaces through menu-based natural language understanding. *Proceedings of the 1983 Conference on Human Factors in Computing Systems*, The Association for Computing Machinery, Inc., New York, 1983, pp. 154-160.
24. Ogden, W.C., and Brooks, S.R., Query languages for the casual user: Exploring the middle ground between formal and natural languages. *Proceedings of the 1983 Conference on Human Factors in Computing Systems*, The Association for Computing Machinery, Inc., New York, 1983, pp. 161-165.
25. Palme, J., A human-computer interface encouraging user growth. *Designing for Human-Computer Communication*, M.E. Sine and M.J. Coombs (eds.), Academic Press, New York, 1983, pp. 136-156.
26. Shneiderman, B., System message design: Guidelines and experimental results. *Directions in Human/Computer Interaction*, A.N. Badre and B. Shneiderman (eds.), Ablex Publishing Co., Norwood, N.J., 1982, pp. 55-78.
27. Badre, A.N., Designing chunks for sequentially displayed information. *Directions in Human/Computer Interaction*, A.N. Badre and B. Shneiderman (eds.), Ablex Publishing Co., Norwood, N.J., 1982, pp. 179-194.
28. Morochove, R., Product comparison: High-end accounting software. *InfoWorld* 10(17):43-55, April 25, 1988.
29. Desposito, J., File managers get a face-lift. *PC Magazine* 6(2):119-133, January 27, 1987.
30. Martin, J.B., Cantrell, M.E., Fichman, R.G., Hunsberger, P.K., and Radoyevich, M.J., An integrated surgical suite management information system. *J. Med. Syst.* 8(6), December 1984.
31. Bullen, C.V., and Rockart, J.F., *A Primer on Critical Success Factors*, Center for Information Systems

- Research (CISR WP 1220-81), Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Mass., June 1981.
32. Rockart, J.F., Chief executives define their own data needs. *Harvard Bus. Rev.* 57(2):81-92, March-April 1979.
  33. Thomas, J.W., and Lowery, J.C., Determining information needs of hospital managers. *Inquiry* 18:300-310, Winter 1981.
  34. Baker, J., SQL: A new standard. *Computerworld* 20(7A):55-58, February 19, 1986.
  35. Babcock, C., Don't Forget SQL's flaws. *Computerworld* 20(37):19, 26-28, September 15, 1986.