Grant No. DSI 77-26524). Philadelphia, School of Library and Information Science, Drexel University, March 1980.

10. Ibid.

11. *Guide to DIALOG Searching*, op. cit., pp. 3-6.

Fox, M.S., and Palay, A.J., (1980), "Machine Assisted Browsing for the Naive User", Public Access to Library Automation, J.L. Divilbiss (Ed.), pp. 77-98, Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign.

MARK S. FOX
Research Scientist
The Robotics Institute
Carnegie-Mellon University

ANDREW J. PALAY
Research Assistant
Computer Science Department
Carnegie-Mellon University

Machine-Assisted Browsing for the Naïve User

Introduction

The purpose of this paper is to demonstrate how a radically different approach to the storage and retrieval of information can result in: (1) a reduction in the need for user sophistication in the use of information systems, and (2) the support of a browsing approach to information system searching. Our approach promotes the view that information system databases should be structured for people, not machines. Many of the problems associated with information systems occur precisely at the interface between the user and information storage. The purpose of the interface is to map user requests onto the database structure. The more "machine-like" the interface, the faster the mapping. As interfaces become more sophisticated (i.e., allow the user to express requests in a more natural form such as natural language), more processing is required to carry out the mapping. And as processing time increases, certain types of search processes (e.g., browsing) become increasingly difficult to provide at a reasonable response rate. Until the time when processing power can meet the real-time needs of system users, information system design and construction should conform to user needs, meaning that the physical structure should be tailored to the user's view (logical structure) of the database, reducing mapping and search time.

In the following discussion, we will describe the BROWSE system, a database browsing system for computer-naïve users.[1] The primary application of the BROWSE system is to allow browsing access to the Carnegie-Mellon University Computer Science Department library collection, but it can be used for other applications, such as the automated dictionary.[2]

We will begin by characterizing our view of the naïve user. Browsing will then be defined, followed by a detailed example of browsing in the BROWSE system. This is followed by an evaluation of BROWSE as a browsing tool for the naïve user. Lastly, we describe the BROWSE system software used to generate browsable information systems.

Characterizing the Naïve User for System Design

A common misconception is that the naïve user of an information system has little or no computer-related experience. Such is not necessarily the case. Naïveté can take many forms:

Computer Naïveté: The person has no experience with a computer. "Intelligent" interaction with a machine is an alien concept. The individual may be reluctant or apprehensive about approaching information systems.
Interface Naïveté: Many information system interfaces are complex and idiosyncratic, requiring many hours of training. New users are unable to use such systems. Information specialists act as intermediaries. Research in natural language understanding is an attempt to make the interface more comfortable.[3]
Information Structure Naïveté: How information is structured in the database, i.e., *what* the categorization hierarchies, is often different from the user's view of the structure. Hence, the ability to specify an information request does not imply that anything will be found.
Category Naïveté: Given that the user does not possess any of the above naïvities, information access can still be hampered. How information is categorized can vary among categorizers. Even if the user knows the categorization hierarchy, the category a user thinks suitable for an entry may not be the one chosen by the categorizer.

Any of these can impede the successful acquisition of information. Hence, information systems must be designed to remove these impediments.

Ignoring computer naïveté, we propose three principles for the design of user-accessible information systems: (1) principle of interface perspicuity, (2) principle of structure perspicuity, and (3) principle of category perspicuity. These principles directly correspond to the last three naïveties listed above. They state that the interface, database structure, and categorization methods must be either apparent to the user or easily learned. Hence, any information system design must account for these principles if it is to be used by naïve users.

Just as important as defining these principles are the methodologies for measuring adherence to them. In measuring interface perspicuity, Card and others have identified several interface performance factors:

Time: How long does it take a user to accomplish a given set of tasks using the system?
Errors: How many errors does a user make and how serious are they?
Learning: How long does it take a naïve user to learn how to use the system in order to do a given set of tasks?
Functionality: What range of tasks can a user do with the system?
Recall: How easy it is for a user to recall how to use the system on a task that he has not done for some time?
Concentration: How many things does a user have to keep in mind while using the system?
Fatigue: How tired do users get when they use the system for extended periods?[4]

An interface should minimize time, errors, learning, and fatigue while maximizing functionality, recall and concentration. What these actual values should be depends on the application. Structure and categorization perspicuity can be measured by experiment: do users find the information they want? Bates has done such an analysis, showing that only 33 percent of library searchers who thought they were successful actually were.[5] How well BROWSE satisfies these factors is discussed later in the paper.

Search Methods: Browsing v. Parameterized Search

The majority of research in the area of database access has focused on the area of parameterized search (PS). PS can be characterized as strictly focused, in the sense that the user must specify exactly the set of attributes that the records must have, e.g., "Get me all the records for items written by Fox about learning." Besides the naïveté problems discussed above, PS systems do not allow quick and easy access to related records. In order to access related material, a new set of search parameters must be specified. Access to related information is the essence of browsing.

Browsing can be characterized as a heuristic search in a well-connected space of records. In particular, browsing can be viewed as an iterative, five-step process:

1. Choose a browsing attribute, such as a category, author, keyword, etc.
2. Access and peruse entries via the chosen attribute (e.g., books, technical reports, etc.).
3. Narrow perusal (search) to small subset.
4. Examine a small subset of entries to confirm interest and find new information.
5. If an entry suggests a new search attribute, then go to step 1 or else go to step 2.

Each of the browsing steps described above is important and is just barely supported in a library. That is, only the author and category attributes are indexed, and distances between shelves impede searches. Both libraries and information systems must provide the ability to search and examine. Even more important, many search attributes must be accessible and new attributes easily searched. Some of the search heuristics most commonly used in browsing library databases are:

1. If book *x* is interesting, then what else has the author of *x* written?
2. If book *x* is interesting, then what other books are in the same category?
3. If a symposium article is interesting, then what else appeared in the same symposium?
4. If the author of an interesting paper is from an institution *x*, then what else has been published at that institution?
5. If there is an interesting paper in a journal, then what else appeared in that journal?

The goal of the BROWSE system is to provide browsing access to databases by building the search heuristics directly into the database as quick-access paths between related records. This includes: (1) providing a simple man-machine interface that takes little training to master, (2) presenting the system in such a way that its logical structure is easily and quickly understood, and (3) providing a browsing-style approach to database searching.

An Example

A BROWSE system database consists of a set of frames. A frame is a single CRT (television) screen of information. The purpose of the frame is to provide information to the user, and to provide quick access to further, related information. Options provide links to related frames. A user moves between the frames by selecting an option, which results in the display of a new frame. Figure 1 is an example from a BROWSE database developed for an on-line library catalog system.

The first frame (fig. 1) welcomes the user to the BROWSE system. In the upper right-hand corner of every frame is the name of the frame (in this case, ZOG1). At the bottom of the frame is a set of standard options (*help, back, next,..., find*). These options, called global pads, will appear in every frame. They provide a set of system functions that are useful throughout the entire network. The first frame consists of the text welcoming the user to the BROWSE system and three options. The first option allows the user to continue receiving instruction on how to use the BROWSE system. The second option allows the user to move directly to the top of the classification hierarchy. The third option allows the user to get news describing changes to the system. To select an option, the user types in the first letter of the option ("I" to select the first option, "T" for the second, "N" for the third). If there is a pointing device (mouse, touch screen, etc.) available, then the user need only point to the option in order to select it. The naïve user would continue by selecting the "I" option for more instruction. An experienced user would select option T, thus displaying the top of the classification hierarchy (fig. 2). There are twelve options to this frame. Options that contain a dash (—) after the first two characters do not point to any other frames. They are included as pointers to information which one day will be included. An important property of the BROWSE system is that it can support multiple views of the database. At present, only the Computing Review classification (1976) and a new entries list are available.

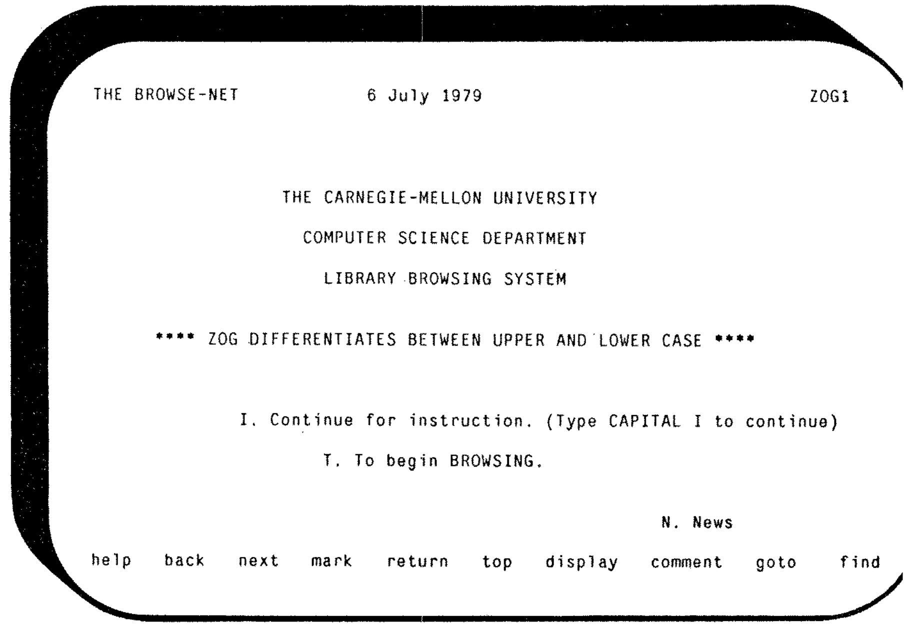


Figure 1

By selecting option 1, the user moves on to the top of the Computing Review classification hierarchy (fig. 3). The frames forming the classification hierarchy each contain a title (0: Computer Science), a definition section, a list of subcategories, an "entry list" option (E), a "lost map" option (L), and a "parameterized search" option (S). If there are additional

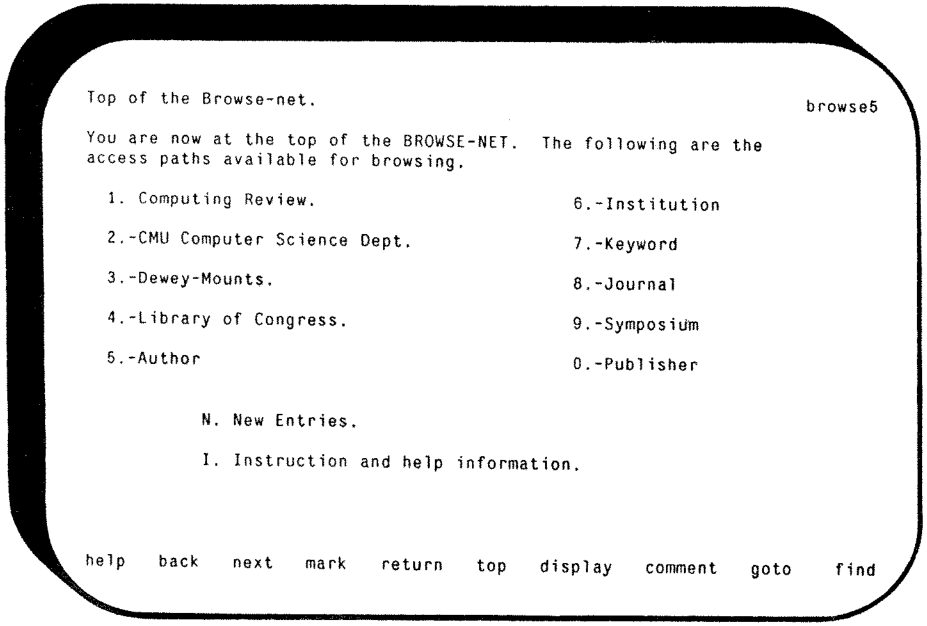


Figure 2

subcategories, then a "more selections" option is included (M). Finally, if there is a designated primary supracategory, then an option (P) is included that links the current classification frame to its primary supracategory. The entry list option points to a list of the entries that have been directly classified under the current category. In this case there are no such entries. Finally, the parameterized search option causes the system to begin a parameterized search and proceed to a set of frames that allows the user to specify the parameters.

The initial goal of this sample session is to browse through the database for information related to learning. By selecting the option for Applications, the user learns that the category includes cognitive processes (fig. 4). By selecting the "more selections" option (M), the user discovers that Artificial Intelligence is one of the subcategories of Applications (fig. 5). If Applications was not a useful category, the P option could be selected to go back up the hierarchy, to allow the user to select another search path.

By selecting option 2, the user moves to the frame describing Artificial Intelligence (fig. 6) and discovers that learning is one of the subcategories (Learning and Adaptive Systems). Note that an entry frame exists for this category. If the user wanted to see entries directly classified under Artificial

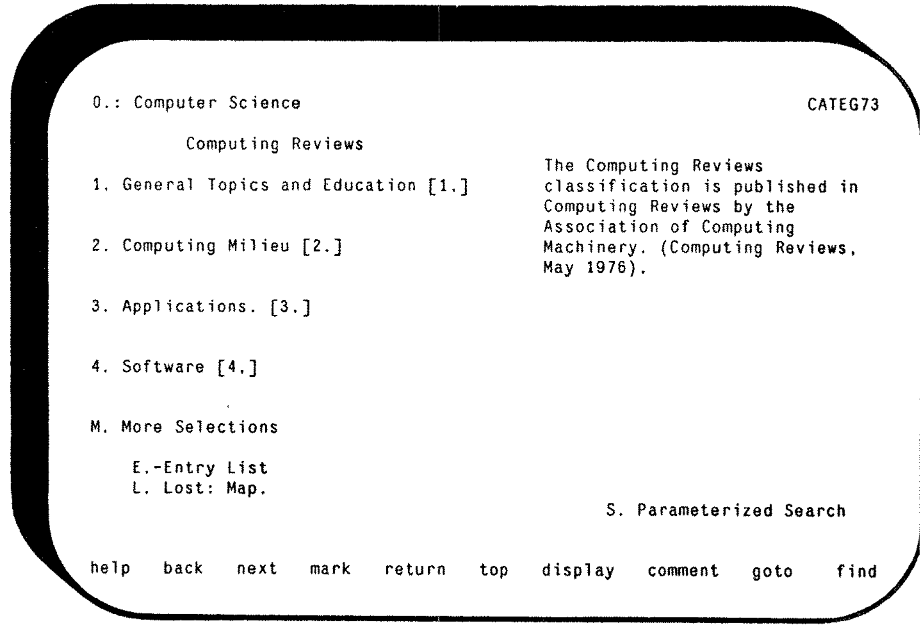


Figure 3

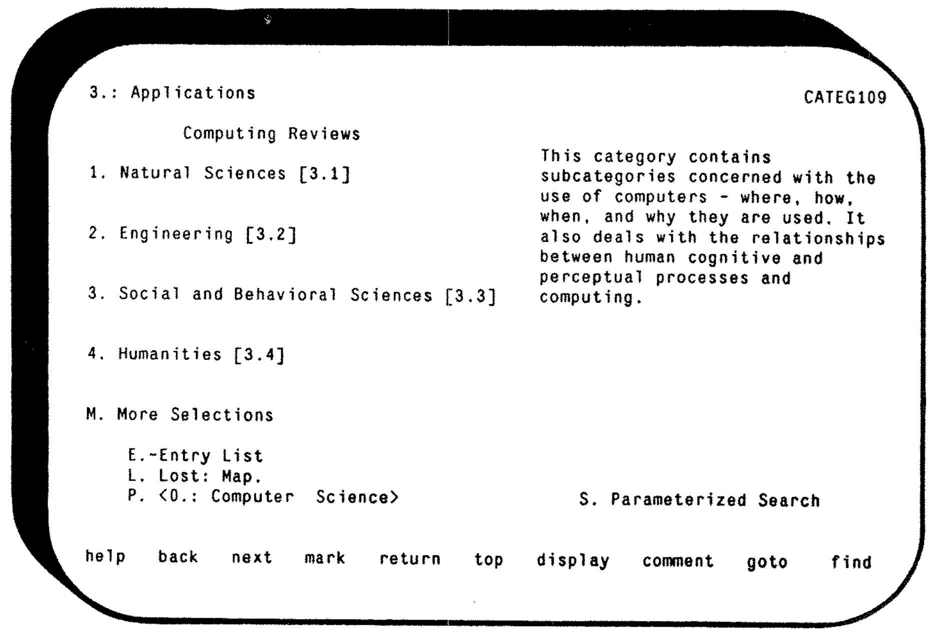


Figure 4

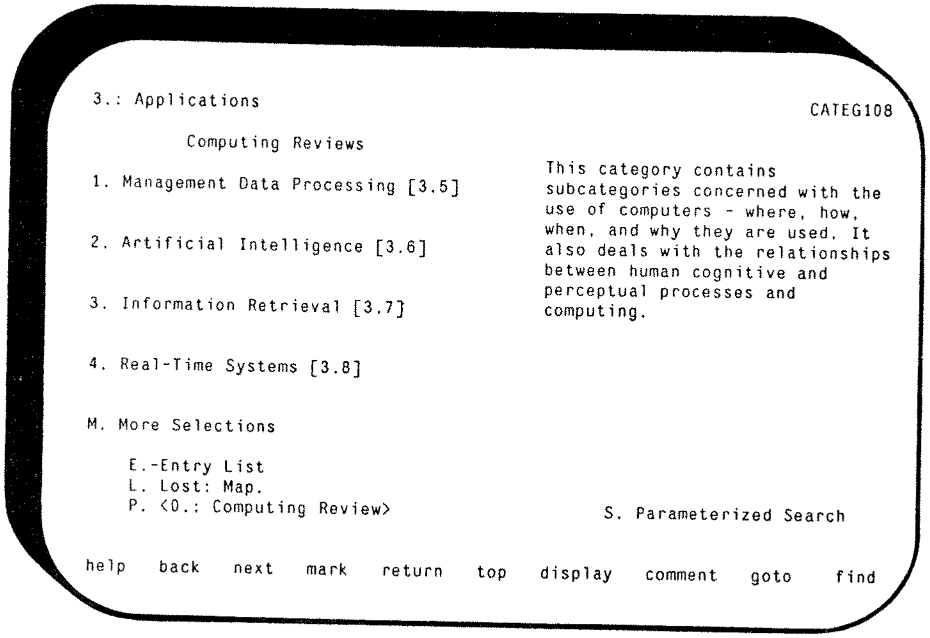


Figure 5

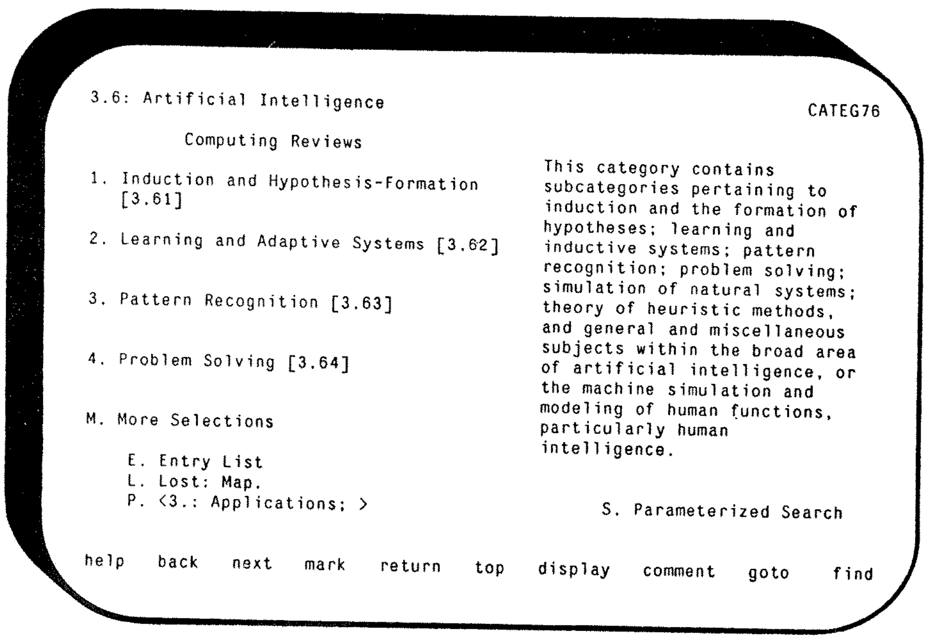


Figure 6

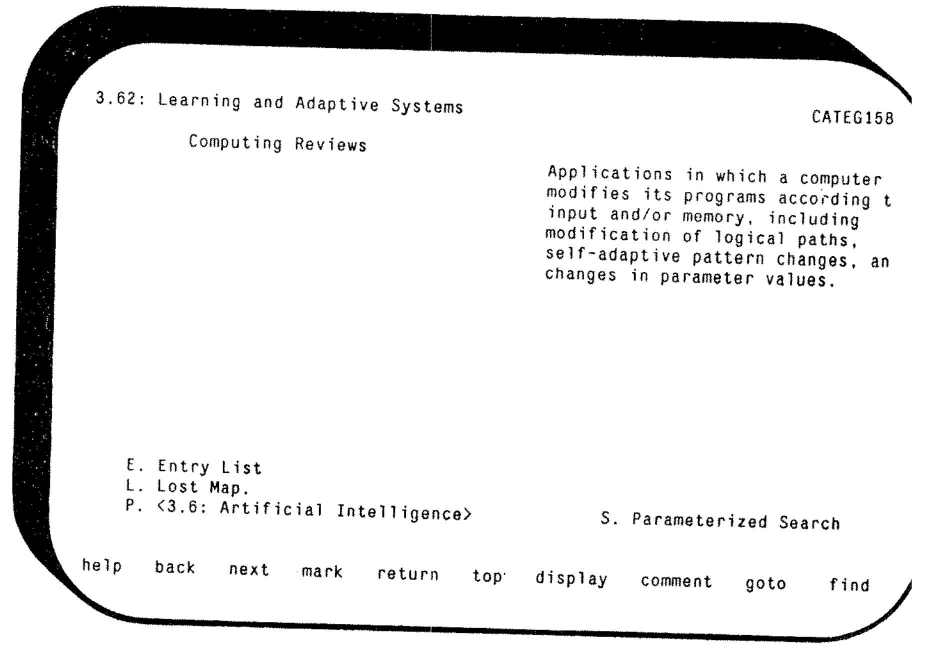
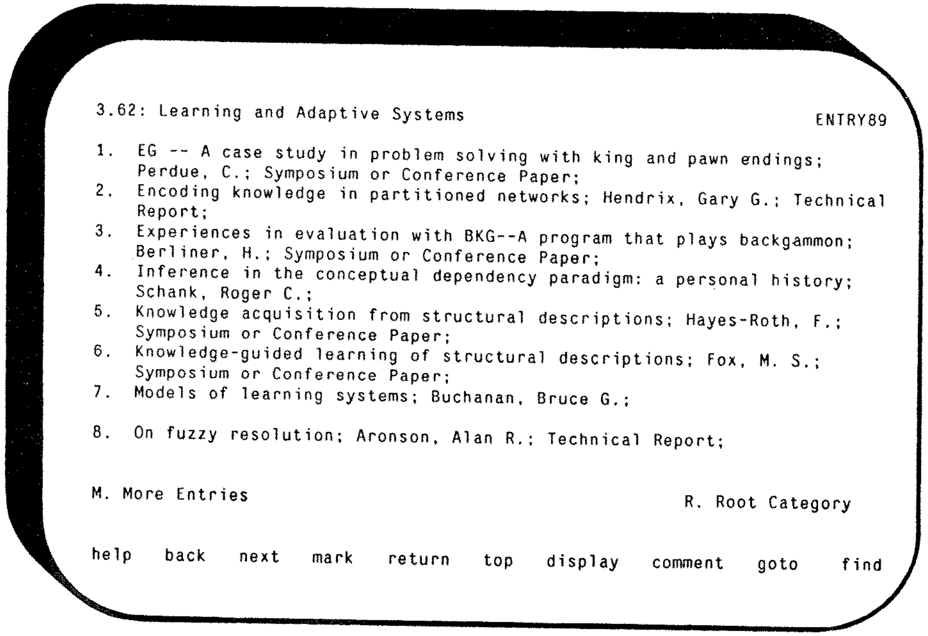


Figure 7

Intelligence, he would select option E. With an interest in learning systems, the user selects option 2, and the goal category frame has been reached (fig. 7).

There are no subcategories to Learning and Adaptive Systems. The user has reached a terminal frame in the classification hierarchy. The entry list (fig. 8) is displayed by selecting the E option. Along with the list of entries, this frame contains a pointer back to the category frame leading to the entry list (option R), and options to move back and forth through the entry list (option M and option P, which is not shown). This is just one form of indexing provided by BROWSE. In addition, the autogeneration of hierarchical and alphabetical indexes has been added.

The user decides that entry 6 looks interesting. Selecting that option, the frame in figure 9 is displayed. This frame provides the basic information about the article in question. The user may find additional information about the authors by selecting options A or B. If the user wants information about the Computer Science Department at Carnegie-Mellon University, he may get it by selecting option I. In all three cases, a list of entries associated with the author or institution will be available. The user can gain additional information about the symposim in which this article



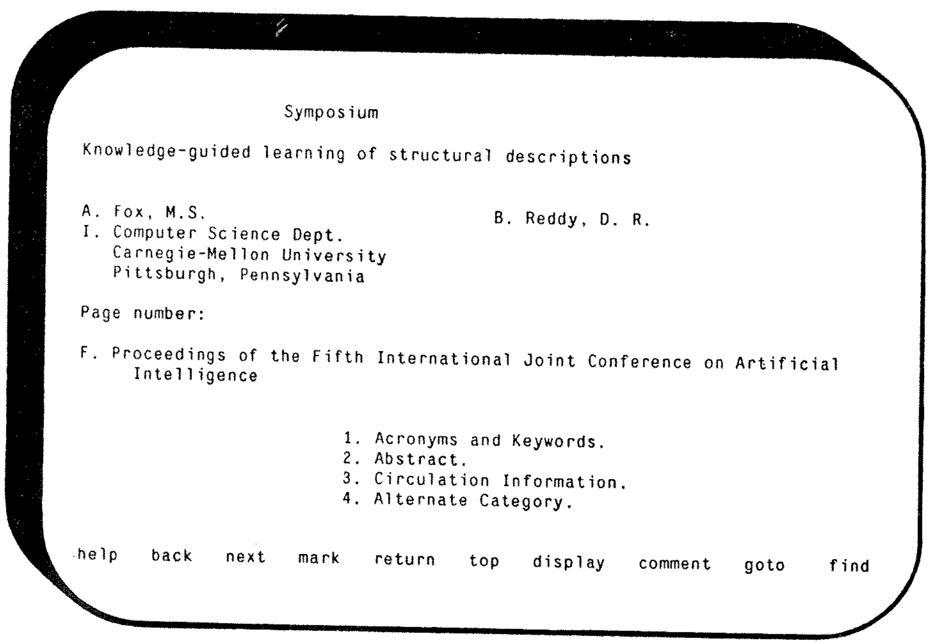
3.62: Learning and Adaptive Systems ENTRY89

1. EG -- A case study in problem solving with king and pawn endings; Perdue, C.; Symposium or Conference Paper;
2. Encoding knowledge in partitioned networks; Hendrix, Gary G.; Technical Report;
3. Experiences in evaluation with BKG--A program that plays backgammon; Berliner, H.; Symposium or Conference Paper;
4. Inference in the conceptual dependency paradigm: a personal history; Schank, Roger C.;
5. Knowledge acquisition from structural descriptions; Hayes-Roth, F.; Symposium or Conference Paper;
6. Knowledge-guided learning of structural descriptions; Fox, M. S.; Symposium or Conference Paper;
7. Models of learning systems; Buchanan, Bruce G.;
8. On fuzzy resolution; Aronson, Alan R.; Technical Report;

M. More Entries R. Root Category

help back next mark return top display comment goto find

Figure 8



Symposium

Knowledge-guided learning of structural descriptions

A. Fox, M.S. B. Reddy, D. R.
I. Computer Science Dept.
Carnegie-Mellon University
Pittsburgh, Pennsylvania

Page number:

F. Proceedings of the Fifth International Joint Conference on Artificial Intelligence

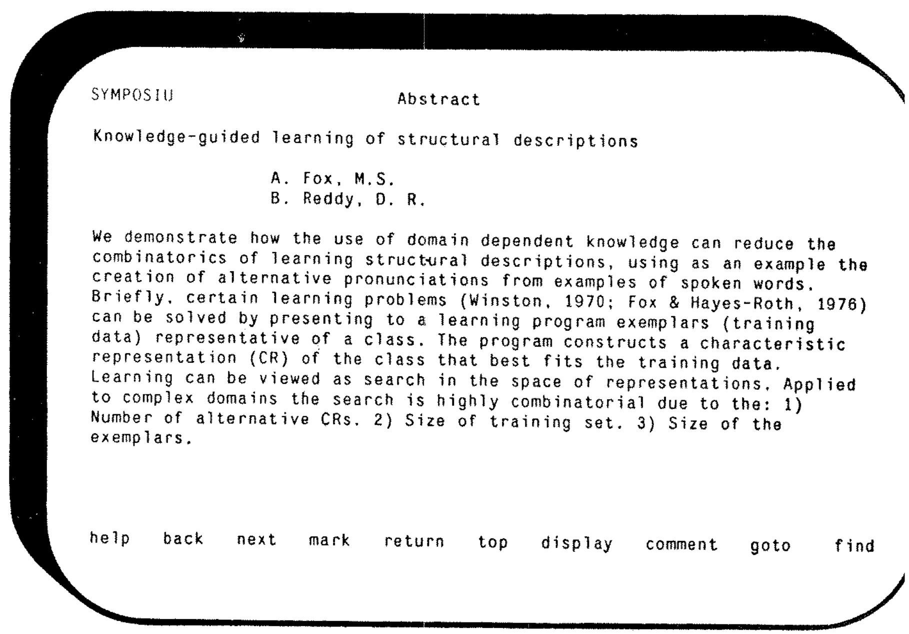
1. Acronyms and Keywords.
2. Abstract.
3. Circulation Information.
4. Alternate Category.

help back next mark return top display comment goto find

Figure 9

appeared by selecting option F. Along with information about the symposium, a list of all articles will be provided. Finally, there is a set of options that provides additional information about the current entry. Option 1 will provide a list of acronyms and keywords. The keywords are organized as a list of options. If the user wants to see what other entries in the database share a keyword, he can select that keyword. The second option provides the abstract to the paper. The third option provides circulation information about the entry in the library. The fourth option leads to a list of categories under which the current article has been classified. These categories are options that point back into the classification hierarchy.

By selecting option 2, the user can view the abstract of the paper (fig. 10). The user decides that he wants to see additional information on M.S. Fox. By selecting option A in figure 10, he moves to a frame giving some information about the author (fig. 11). Options are provided for linearly moving through the author list (<,>) and going to the author index frame (↑). By selecting option 1, the user is led to a frame listing all of Fox's articles that are currently in the database (fig. 12). Seeing nothing of additional interest, the user now decides to see Fox's other areas of interest (fig. 13).



SYMPOSIU Abstract

Knowledge-guided learning of structural descriptions

A. Fox, M.S.
B. Reddy, D. R.

We demonstrate how the use of domain dependent knowledge can reduce the combinatorics of learning structural descriptions, using as an example the creation of alternative pronunciations from examples of spoken words. Briefly, certain learning problems (Winston, 1970; Fox & Hayes-Roth, 1976) can be solved by presenting to a learning program exemplars (training data) representative of a class. The program constructs a characteristic representation (CR) of the class that best fits the training data. Learning can be viewed as search in the space of representations. Applied to complex domains the search is highly combinatorial due to the: 1) Number of alternative CRs. 2) Size of training set. 3) Size of the exemplars.

help back next mark return top display comment goto find

Figure 10

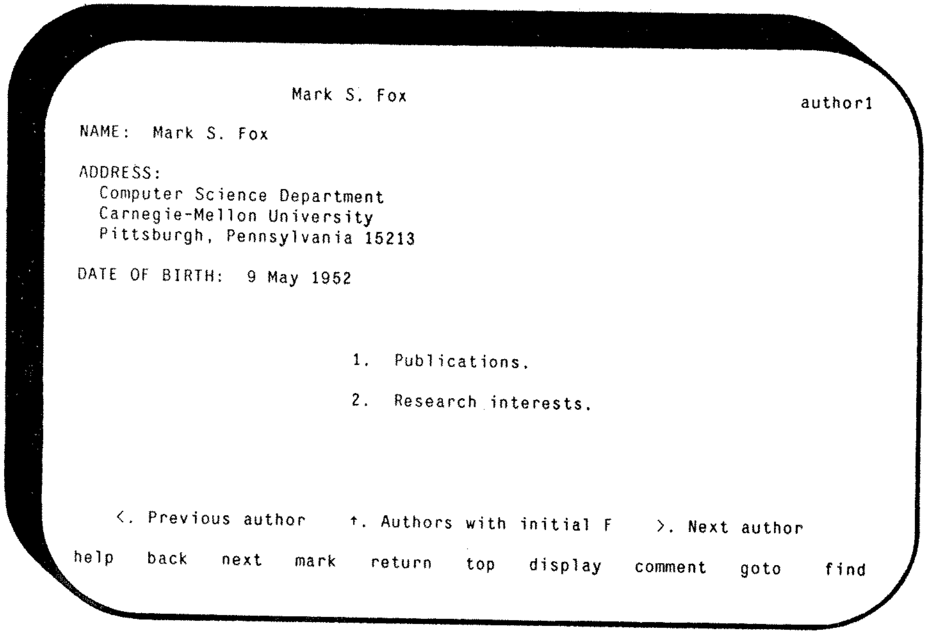


Figure 11

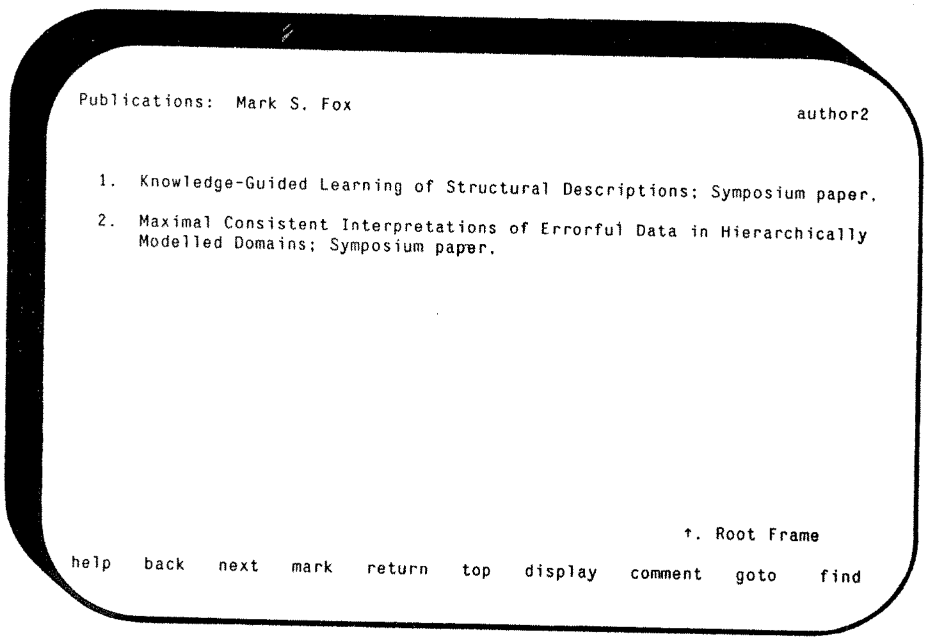


Figure 12

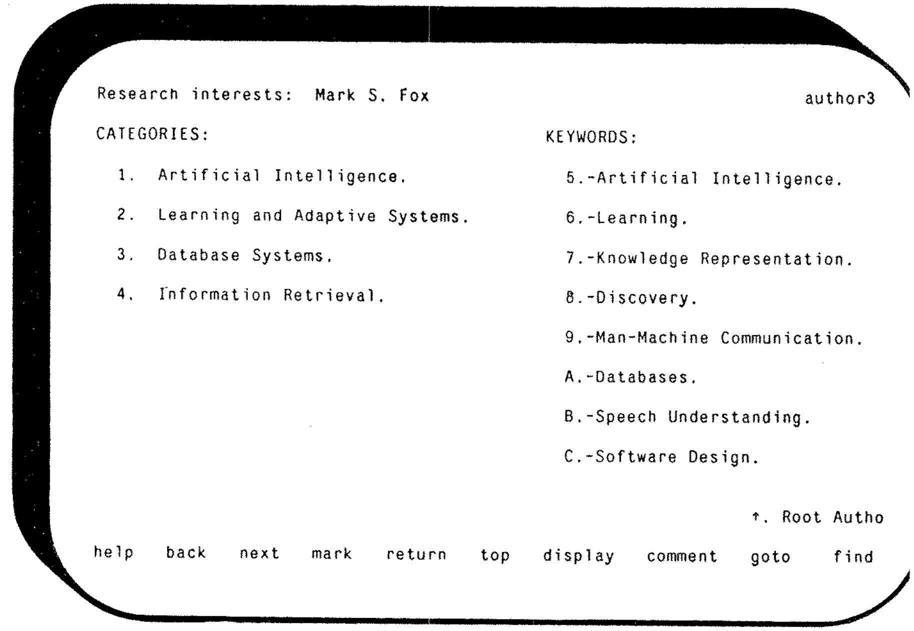


Figure 13

Also interested in information retrieval, the user selects option 4. This puts him back into the classification hierarchy (fig. 14). He may now continue browsing in that area. If the user has managed to get lost, there is the "lost map" option L. By selecting that option, the user can get a global view of the classification hierarchy that surrounds the current category (fig. 15). Surrounding categories can be reached by selecting any of the options.

Browsing and Searching

The BROWSE system relies on browsing as its primary method of database access. However, there are many times when parameterized search is desired. The user may already know exactly what he is looking for and should not have to move through the network of frames to get there.

The BROWSE system includes the ability to specify searches. As was shown in figure 2, each category frame in the current system has an option that allows the user to specify a parameterized search. The parameterized search differs from searches in normal PS-based systems in the following ways:

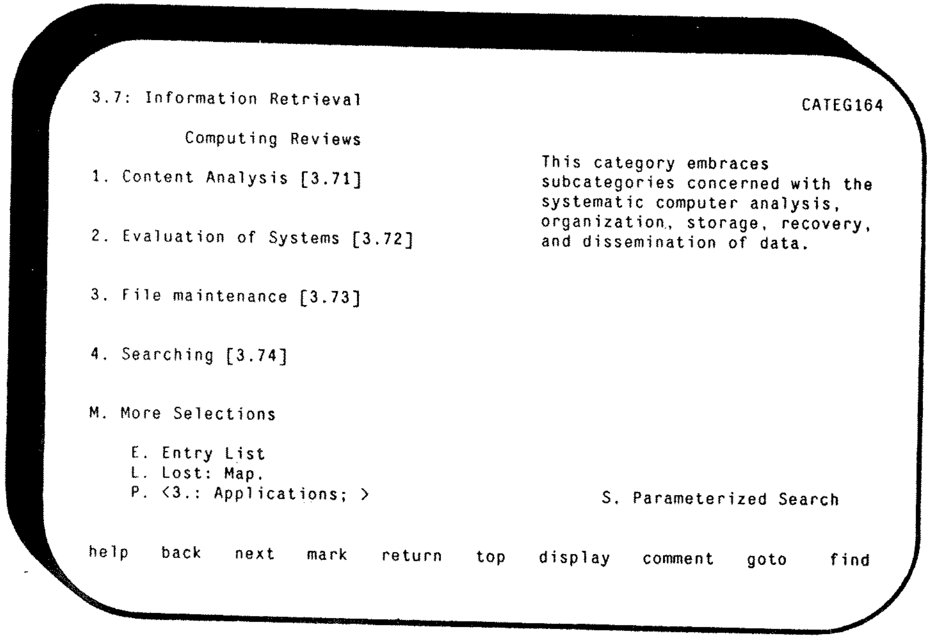


Figure 14

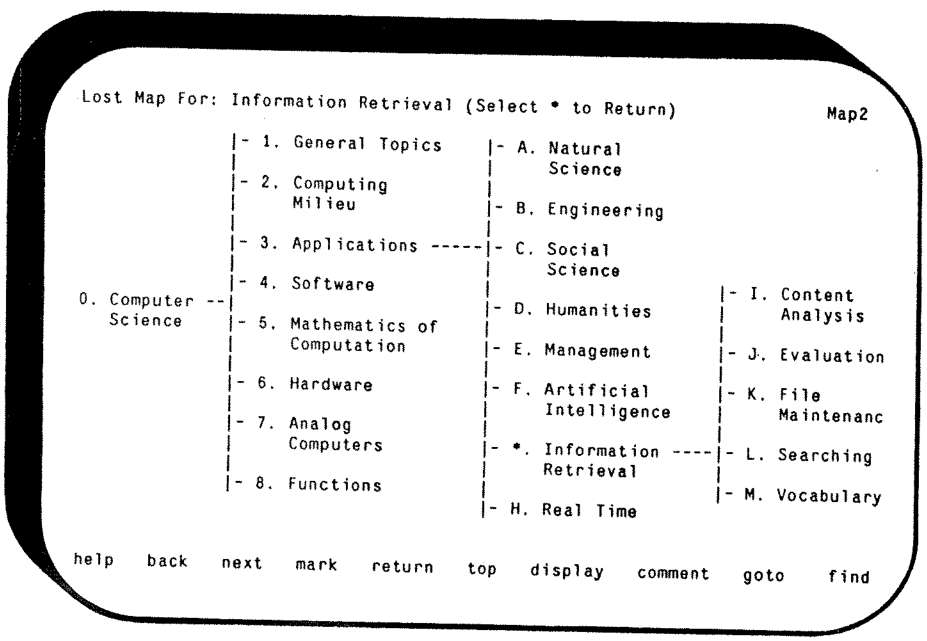


Figure 15

1. The search that is initiated is context-dependent. The user may browse through the classification hierarchy until he finds the area in which he is interested. When he initiates a search at that point, only entries classified under that category will be searched. Thus, if the user selected the "parameterized search" option at the frame for Artificial Intelligence, the procedure would only search entries directly classified under Artificial Intelligence and entries classified under all of the subcategories to Artificial Intelligence.
2. When the results of the search are returned to the user, he is able to browse through the list of found entries. He may also browse outside the list of satisfied entries (follow any link). The current search procedure returns a list of frames in the system that satisfy the search parameters. An additional set of options are provided that allow the user to move through the list. The list can be viewed as an additional classification structure. Thus, the user can make full use of the browsing capacity of the system.

The ability to combine browsing and searching reduces the problem with parameterized searches of not finding all the entries in which the user is interested. Often, closely related entries are not returned by the search. If the user is allowed to browse throughout the network, he can often find the entries related to the list of returned entries.

Measuring the Efficacy of the BROWSE System

We have shown what using the BROWSE system is like. Its purpose is to provide naïve users browsing access to information. Its success rests on how well it adheres to the principles of interface, structure and category perspicuity. Consider the interface factors described earlier:

1. *Time*: The search heuristics described earlier are the primary functions. They are provided by selecting an option in order to transfer to another frame. Frame change is rapid. Hence, the time it takes to accomplish a browsing task is small. The response time of parameterized search depends on the machine, the operating system and the program.
2. *Errors*: Restricting control to option selection virtually eliminates the problem of incorrect commands. When a user makes an incorrect option selection, the "back," "return" and "goto" global pads provide sufficient recovery. If the user gets lost, the lost map frame provides sufficient context.
3. *Learning*: The single most important control construct to be learned is option selection. This is a simple task to learn. Any other instruction is provided by frame text.

4. *Functionality*: Search heuristics are embedded in options. Other functions such as parameterized search are also provided by options.
5. *Recall*: The primary command to be remembered is option selection. Instruction is usually provided by the text of each frame. And the "help" global pad provides access to general instruction frames from every frame.
6. *Concentration*: The concentration required of the user is dependent upon the task. This example use of ZOG records the sequence of frames visited by the user so that the list may be displayed and reviewed. Nevertheless, complex searches may require the user to remember the nonlinear topology of the subnet visited. The difficulty of remembering spatial orientation results in the feeling of being lost. The problem is being studied,[6] and a number of approaches for dealing with it include the dynamic construction and display of maps of subnets visited by the user.[7]
7. *Fatigue*: Rapid response results in the system waiting for the user, not vice versa. Hence, the user does not tire from waiting for results.

Structure perspicuity is maintained in three ways: (1) each category lists its subcategories and also has a link to its supracategories, thus displaying a part of the categorization structure to the user; (2) the user can move through the structure quickly and easily, allowing assimilation of the structure; and (3) the "lost map" option (fig. 15) gives a more global view of the categorization structure. Category perspicuity is partially solved by providing a definition on each category frame. The user can also learn categories by example; he can quickly access all the entries in a particular category. As a system for naïve users, BROWSE appears to satisfy the principles well. Coupled with parameterized search capability, it provides a powerful system for public access to information.

Building BROWSE Systems

The BROWSE system is composed of two separate systems. There is a display system called ZOG and a system that is used to create the frame-structured database called the BROWSE System Software.

The ZOG System

The BROWSE system was designed to use ZOG, developed at Carnegie-Mellon University by Robertson and others, as its display system.[8] ZOG has its roots in the University of Vermont's PROMIS system.[9] To understand the philosophy of BROWSE, one must understand the philosophy of ZOG. ZOG is a rapid-response, large-network, menu-selection system for man-machine communication. A ZOG user sits in

front of a terminal on which a frame is displayed. The frame consists of some text and a set of options. At the discretion of the user, an option is selected and almost instantaneously a new frame is displayed. The process then starts again. ZOG's basic features are:

1. *Rapid response*: When a user selects an option that leads to another frame, the next frame should appear fast enough so that the user does not feel he is waiting for the system. A user must feel free to explore surrounding frames without concerning himself with the time it takes to display each frame.
2. *Simple selection*: The act of selection should be a simple unitary gesture. At the present time, there are two forms of selection available: (1) single character input from the keyboard, and (2) a touch screen.
3. *Large network*: The network should be large enough to provide most of the information needed by the user.
4. *Frame simplicity*: The frame display should be kept simple. The user should be able to assimilate the information contained in the frame quickly. The idea of frame simplicity has led to the development of frames that contain a small amount of text and up to five or six options. This is not necessarily true of frames developed for the BROWSE system, where the simplicity comes from the very structured nature of the information. Although a large amount of information may be displayed on a single frame, the information desired by the user can still be assimilated quickly.
5. *Transparency*: The user should be able to understand exactly what the system is doing and what he needs to do to gain additional information. At no point should the user feel that he has lost control of the system.
6. *Communication agent*: ZOG has been designed to act as a communication agent between a user and another system. As a communication agent, ZOG presents commands to the user in a simple format, as well as an explanation of what the command will do. When the user makes a selection, ZOG sends the more complex set of commands to the other system for processing. This facility is used by the BROWSE system for the parameterized search interface.
7. *External definition*: Unlike many menu-selection systems, ZOG-nets are databases which are independent of and external to the ZOG system.

The basic philosophy of ZOG is that a menu-selection system can be an effective communication system if the user can move around in the system quickly, and if there is a large network available to meet the user's needs.

A menu-selection system allows the user almost complete knowledge of what is occurring in the system whenever he selects an option. It also allows for related information to be located nearby (by placing a link between related frames). Menu-selection systems normally have a disadvantage in the time it takes to move from one frame to another. This is solved by the fast response of the ZOG system. Another characteristic of many menu-selection systems is that the same information is provided to all users, regardless of their needs or expertise. In ZOG, different paths would be provided for each level of user. The naïve user would get more information about what he is doing, while the expert user would be presented with only the frames needed to perform the task.

The BROWSE System Software

A major problem with menu-based systems is the creation of the menus. Experiments with the ZOG system have shown that the average rate of frame creation is approximately five frames per hour.[10] Given a database that contains in excess of 1 million frames, it becomes clear that some machine-aided creation mechanism is needed.

The BROWSE system solution to that problem was to create a software package external to the display system (ZOG) to create the database network. The software is designed to translate a database into a frame database. The reasons for using an external database and a translation system are:

1. To allow for the modification of the frame formats. The design of frame formats and network structures may change, requiring the frame database to be recreated. ZOG does not provide facilities for the automatic modification of frames and structure.
2. To allow for the creation of different networks for different display types. At the present time the BROWSE system runs only on a standard 24-by-80-character display terminal. A completely different frame structure would be used for a high-resolution display terminal.
3. To allow for parameterized searches. It would be cumbersome to carry out the parameterized search within the frame network. Thus, the search is carried out by a separate program that accesses the original database.

The BROWSE system is a set of software that provides the following capabilities: the ability to create a record database and new record types with access and search software; interactive programs to query data-entry personnel for new entries for the database; interactive definition of frame formats; interactive definition of record linkages which define how frames are linked in the frame network; and frame creation, by combining: (1) a new record in the database, (2) a frame format description, and (3) a linkage specification; to produce new frames and updated frames (e.g., index frames), with the proper option links, for the frame database. Hence, the

total effect of the software is to translate a record database into a browsable frame database for naïve users.

Recognizing that the visual format of frames is important to the overall acceptability of the system, we designed the frame format description language to provide a variety of layout capabilities. The frame format describes:

1. the information to be placed in the frame (fields of a record),
2. the placement of the information in windows on the screen,
3. the information actually necessary to create the frame,
4. the options to appear in the frame, and
5. the type of frame to which each option may lead.

The frame format consists of (in increasing order of complexity): (1) window descriptions, (2) option descriptions, (3) fill descriptions, (4) group descriptions, and (5) index descriptions.

The window description (WD) is the basic unit of the frame format. The WD includes information describing a two-dimensional area into which text will be placed. The WD contains a starting coordinate for the window. This position is relative to the group that accesses the window. The WD also contains a minimum and maximum length and width of the window.

The option description (OD) contains information that relates to options in the ZOG frame. Included in that information is the selection character for the option, the touch area of the option (in case a pointing device is available), the text to be included as part of the option text, the frame type to which the option leads, and a WD that is to be used to hold the text of the option.

The fill description (FD) describes the information (fields) from the database that is to be used to fill a window. The FD also points to a WD. In evaluating an FD, the system retrieves the contents of the fields in the current record. The FD contains information on what to do if the contents of a field are null, as well as the text to place before and after the text found. All the text specified is concatenated and then placed in the window. Also included in the FD is information describing the size, typeface and font to be used in displaying the text (for use with a high-resolution display), as well as commands describing how to display the text in the window (centered, flush right, flush left, filled).

The group description (GD) is one of the two major units of the frame format. A GD combines a set of FDs (and possibly index description, or an ID) into a single logical unit. The GD points to a set of FDs. If the GD is describing an option, then the GD will also point to an OD. The GD also contains an absolute anchor point for the group. An anchor defines the root position of the group on the frame. The starting positions provided by the WDs and referenced by the GD are relative to that anchor point. The anchor point of the group can be set relative to other groups (i.e., the anchor point of group X can be placed one line below the last line used by group Y).

The other major unit of the frame format is the index description (ID). Associated with every category frame is an option to a list of entries. That list is a linear index, ordered alphabetically by title, to all the entries in that category. One of the major problems with an evolving system is that the indexes to the system must constantly be rebuilt. The ID has been included in the frame format to allow the system to create indexes mechanically. The system is capable of creating a variety of index types (linear, hierarchical, alphabetical). The ID points to two sets of GDs. The first set refers to groups that are used as titles to the index. The second set is used for actually creating the index. The ID also contains information on how to form the list of entries to be used in creating the index.

Conclusion

The application of computer technology to information and library systems has created a vista of opportunities. The possibility of a revolution in how information is stored, accessed and manipulated beckoned, but the actual introduction of technology produced as many problems as it solved. These systems are difficult to learn and use, creating a serious barrier to the naïve user. Second, some researchers were clouded in their thinking by the way they used information systems. They were unable to see beyond their current horizon, thus asserting that computers could not provide certain styles of interaction such as browsing.[11] The main effect of this was a plethora of research in the parameterized mode of search. The combination of the BROWSE and ZOG systems provides a radically different approach to the access and display of information. BROWSE and ZOG provide an integrated browsing and parameterized approach to searching databases, while utilizing an interface that is simple and clear for even the naïve user.

ACKNOWLEDGMENTS

This research was sponsored by the Office of Naval Research under contract No. N00014-76-0874, and, in part, by the Defense Advanced Research Projects Agency (DOD), Arpa Order No. 3597, monitored by the Air Force Avionics Laboratory Contract 533615-78-C-1551. The authors wish to thank the people involved in the ZOG project for providing the important display component of the BROWSE system; also, special thanks to Earl Mounts, head of the Computer Science Browsing Room at Carnegie-Mellon University, for his help in developing the BROWSE system.

REFERENCES

1. Fox, Mark S., and Palay, Andrew J. "The BROWSE System: An Introduction." *In* Roy D. Tally and Ronald R. Dueltgen, eds. *Information Choices and Policies: Proceedings of the ASIS Annual Meeting*. Vol. 16. White Plains, N.Y., Knowledge Industry, 1979, pp. 183-93; and ________. "The BROWSE System: Phase II and Future Research." Pittsburgh, Pa., Computer Science Dept., Carnegie-Mellon University, 1979. (unpublished)

2. Fox, Mark S., et al. "The Automated Dictionary," *Computer* 13:35-48, July 1980.

3. Mylopolous, J., et al. "TORUS: Natural Language Understanding System for Data Management." In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence* (Tbilisi, USSR, 1975). Cambridge, Mass., Artificial Intelligence Laboratory, 1975 pp. 414-21; Waltz, D. "Natural Language Access to Large Data Bases: An Engineering Approach." In *Proceedings*, op. cit., pp. 868-72; Sacerdoti, E.D. "Language Access to Distributed Data with Error Recovery." In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (Cambridge, Mass., 1977). Pittsburgh, Carnegie-Mellon University, 1977; and Hayes-Roth, F., et al. "Understanding Speech in the Hearsay-II System." *In* Leonard Bolc, ed. *Speech Communication with Computers*. Munich, Macmillan, 1978, pp. 9-42.

4. Card, S.K., et al. "The Keystroke-Level Model for User Performance Time with Interactive Systems" (AIP Memo 122). Palo Alto, Calif., Xerox Palo Alto Research Center, 1979. (Report SSL-79-1)

5. Bates, Marcia J. "Factors Affecting Subject Catalog Search Success," *Journal of the ASIS* 28:161-69, May 1977.

6. Mantei, M. "Disorientation Behavior in Man-Computer Interfaces." Ph.D. diss., Annenberg School of Communications, University of Southern California, 1981.

7. Fox and Palay, "The BROWSE System: Phase II," op. cit.

8. Robertson, G., et al. "The ZOG Approach to Man-Machine Communication" (Technical Report). Carnegie-Mellon University, Dept. of Computer Science, 1979.

9. Shultz, J., et al. "An Initial Operational Problem Oriented Medical Record System—For Storage, Manipulation and Retrieval of Medical Data." In *AFIPS Conference Proceedings: National Computer Conference and Exposition*. Vol. 38. Montvale, N.J., AFIPS Press, 1971, pp. 765-77.

10. McCracken, Donald L., and Robertson, George G. "Editing Tools for ZOG, a Highly Interactive Man-Machine Interface." *In* Institute of Electrical and Electronics Engineers. *ICC '76: International Conference on Communications*. New York, IEEE, 1979, pp. 22.7.1-7.5.

11. Myers, Charles A. *Computers in Knowledge-Based Fields*. Cambridge, Mass., MIT Press, 1970.