AN EVALUATION OF COMPUTER-SUPPORTED BACKTRACKING IN A HIERARCHICAL DATABASE

by

Cortney G. Vargo

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Industrial and Systems Engineering

APPROVED:

Dr. R. C. Williges, Chairperson

Dr. J. C. Woldstad

Dr. C. E. Brown

Blacksburg, Virginia

March, 1992

AN EVALUATION OF COMPUTER-SUPPORTED BACKTRACKING IN A HIERARCHICAL DATABASE

by

Cortney Grady Vargo

Committee Chairman: Robert C. Williges
Industrial and Systems Engineering

(ABSTRACT)

A common concern for people using computer databases is becoming "lost" within the complex hierarchy of entries. Most direct manipulation interface design guidelines suggest designers should include a feature for "undoing" user inputs (Smith and Mosier, 1986). In the case of a database, undo translates to backtracking support. The first purpose of this research was to confirm that computer-supported backtracking tools reduce navigation time over manual backtracking. The second purpose was to compare navigation times among a subset of backtracking tools. The third purpose was to determine if users prefer to use one or more backtracking tools significantly more than others.

Four backtracking tools were developed by crossing two factors: History (*history list* vs *no history list*) and Level (*component* vs *entry*). *History list* indicates the user may view a chronological listing of nodes that have been viewed and directly select a destination node. *No history list* means the user must backtrack through each visited node with no shortcuts. *Component*

indicates the backtracking tools operate only at the lowest level, or smallest definable node, of the tree-like database structure. *Entry* means that backtracking occurs at the higher parent node. Thus, multiple *components* make up an *entry* . In addition to the four computer backtracking tools, overall navigating and manual backtracking was done using a hierarchical Table of Contents.

The tools were evaluated in an experimental, hierarchical, direct-manipulation database. Trials were conducted in the form of a multiple-choice information retrieval task. The independent variables included the backtracking tool (four-computer supported, one-manual) and the backtrack Task Length. The dependent measures included navigation time, the frequency with which the computer tool was used over manual backtracking (Table of Contents), and questionnaire responses.

The results of this study provided some of the first solid support for the many guidelines that have been written recommending user recovery, or undo support. Backtracking with any of the four computer-supported tools resulted in a significantly smaller navigation time than manual backtracking using the Table of Contents. Subjects using either of the entry tools had consistent backtracking times across trials regardless of backtrack task length. When provided with a history list, subjects in the entry condition had significantly smaller navigation times than subjects in the component condition. Users did not show any differences between computer tools in rated efficiency, ease of use, or objective preference measures.

ACKNOWLEDGMENTS

The author wishes to express appreciation to Dr. Robert C. Williges for his expertise and advice while serving as chairman of this committee. Thanks are also due to Dr. Jeffrey Woldstad for his time, effort, and assistance. Tremendous gratitude goes to Dr. Clifford Brown, my friend and mentor. He offered thousands of words of wisdom and countless hours of consultation making this experience truly enjoyable.

This research was made possible through the Air Force Office of Scientific Research summer research program. The author is indebted to many people for help in completing this research: Ken Boff and Don Monk (Human Engineering Division, Armstrong Laboratory, Wright-Patterson AFB), Sarah Swierenga, Becky Donovan, Glenn Johnson, Keith Adams, and Chuck Goodyear (Logicon Technical Services, Inc.).

On a personal note, I would like to thank my parents Ted and Donna, my sister Kristen, and Allie for believing in me and providing encouragement and stability during these years of graduate school.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Computer Aided Systems Human Engineering: Human Performance Visualization Subsystem

The Computer Aided Systems Human Engineering: Human Performance Visualization Subsystem (CASHE:HPVS) is an interactive, hypermedia database that is being constructed to provide simplified access to human engineering information. It is designed for CD-ROM and will run on the Macintosh™ family of computers. It contains information from the *Engineering Data Compendium* by Boff and Lincoln (1988), *MIL-STD-1472D*, and a Perception and Performance Prototyper that offers hypermedia demonstrations of human factors phenomena.

Purpose of CASHE:HPVS. The CASHE:HPVS project is being co-developed by the Armstrong Laboratory Crew Systems Directorate, Human Engineering Division, Design Technology Branch at Wright-Patterson Air Force Base, the Federal Aviation Administration, the Army Human Engineering Laboratory, the Air Force Office of Scientific Research, and the Naval Ocean Systems Center. The main purpose is to facilitate the integration of human engineering data into the man-machine interface design process. Lincoln and Boff (1988, p. 1021) say that "designers face special obstacles in accessing, interpreting, and applying human performance data that tend to discourage their use." They cite three main reasons for these obstacles. First, there are hundreds and thousands of publications in which the needed information may be found. Second, this information is mainly found as experimental results and may not be readily applicable to design. Third, the users may have difficulty in determining the validity or generalizability of the research results. The CASHE:HPVS attempts to narrow the gap between researchers and designers by providing human engineering information that is readily accessible and applicable.

Ease of information retrieval in CASHE:HPVS is largely determined by the degree of usability of the interface. Therefore, a number of issues must be investigated to ensure a simple, understandable front-end for CASHE:HPVS. This particular study was conducted to determine performance and preference trade-offs associated with four navigation alternatives for backtracking through the system's entries. The results of this study will be integrated into CASHE:HPVS; it is therefore necessary to outline specific CASHE:HPVS interface attributes to demonstrate similarities between the experimental database and the CASHE:HPVS prototype.

CASHE:HPVS is being implemented in a direct manipulation environment; this is relatively new, especially for large-scale databases. The construction and manipulation of CASHE:HPVS is somewhat unique and will be discussed in the following sections.

Composition of CASHE:HPVS. The CASHE:HPVS database draws on a metaphor of a bookshelf containing four volumes: (1) the *Engineering Data Compendium,* (2) *MIL-STD-1472D*, (3) the Perception and Performance Prototyper, and (4) Project Files (Boff, Monk, Swierenga, Brown & Cody, 1991). The *Compendium* contains text, figures, tables, and illustrations describing over 70 areas of human factors. *MIL-STD-1472D* contains government specifications for factors pertaining to system interfaces such as displays, controls, and environmental influences. The Perception and Performance Prototyper allows users to manipulate and experience certain human factors phenomenon to enhance their understanding and appreciation. An example of this is a demonstration of how masking affects audible signal detection. Lastly, the Project File notebook contains any personal changes or additions the user may want to make to the database system.

The major building block of CASHE:HPVS is the entry. Both the book and computerized versions of the *Compendium* are based on this structure. Each entry is a singular, stand-alone piece of information that may contain the following sections: title, key terms, general description, applications, methods, experimental results, empirical validation, constraints, key references, and/or cross references. The entry is composed of some or all of the following components: text, table, figure, and prototyper. The entry always has a text component and may or may not have a figure component, table component, or Prototyper component. There may be more than one figure, table, or prototyper component. The database is hierarchical with components (figure, table, etc.) making up the lowest level (children) and the entry name acting as the parent.

Within an entry, the components are accessible via the "BRASS PLATE" section of the interface. The prototype CASHE:HPVS interface including the Brass Plate region is shown in Figure 1. The USER ANNOTATION REGION includes features for customized bookmarks and notes. It also has an option whereby the user may formally establish and disconnect links between entries

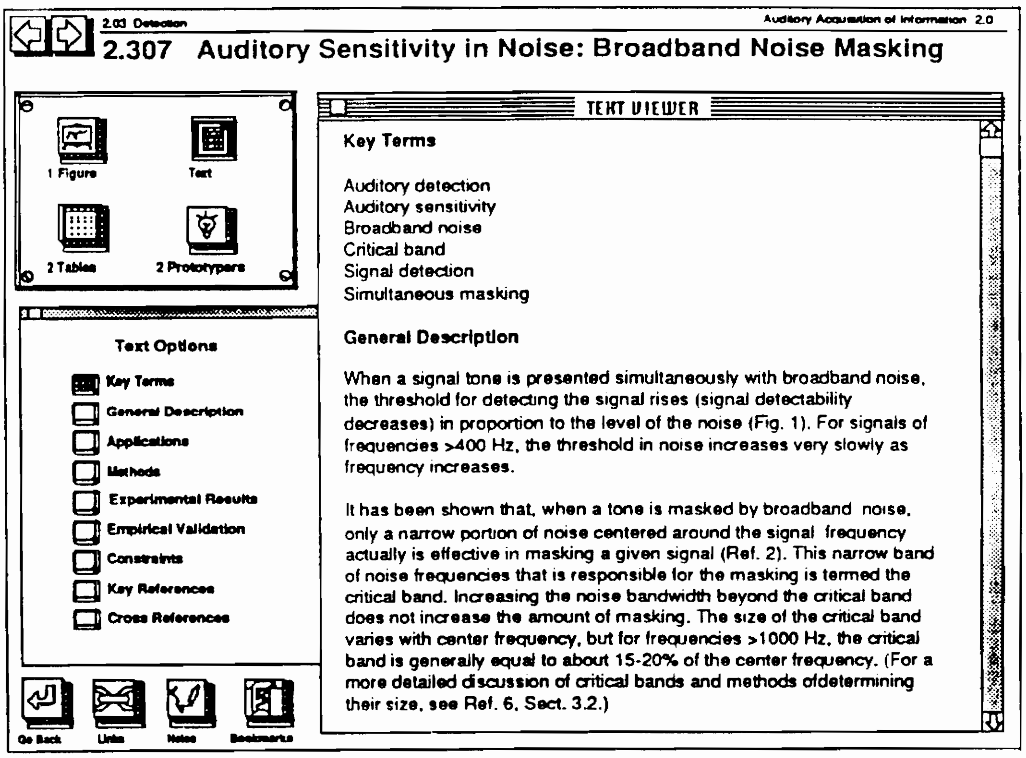


Figure 1. Text component of a sample CASHE:HPVS entry. (taken from Boff et al., 1991)

that may be accessed via a dialog box. The OPTIONS PANEL has controls that may be set by the user and unlike the others, this region contains unique controls for the specific element that is in the ELEMENT VIEWER. However, the other regions (BRASS PLATE, USER ANNOTATION REGION) remain the same throughout the database.

Navigating in CASHE:HPVS. Historically, databases were searched by key words with boolean operators. The database was customarily command- or menu-driven with no simple method for mere "browsing." The CASHE:HPVS system is much more versatile in this respect and provides multiple routes between entries. This is somewhat unique and requires clarification.

To locate a particular entry, there are several options available. If the user knows the proper subject, the user may choose to employ the hierarchical TABLE OF CONTENTS, INDEX, USER TAXONOMIES, or GLOSSARIES. When a key term is known, the user may conduct boolean text searches. If merely browsing, the user navigates through entries numerically using the forward and backward arrows in the upper left hand corner of the display. When several entries are often consulted together, the user may establish hard links between entries through the LINK button. The user may re-activate a background window to re-access a previous entry. Cross references may be used to locate related entries. Lastly, the user may backtrack through entries and/or components using the GO BACK function. This study was conducted to formulate recommendations regarding the functionality of the GO BACK feature in the CASHE:HPVS database.

Rationale

Direct manipulation databases are relatively new. Therefore, no specific guidelines exist for implementing navigation and backtracking systems. Many questions were unanswered regarding the function of GO BACK. The most fundamental question is how necessary is this backtracking feature and when will people use it? Does it reduce the navigation time for users or does it clutter the interface and increase the cognitive workload? The main purpose of the database is information retrieval *not* navigation. Therefore, there are many tradeoffs involved in determining the correct level of complexity and functionality for GO BACK. The backtracking feature could operate at either the entry or component level. The GO BACK feature could display each previous screen sequentially or the user could access a component directly from a history list.

Navigation tools in a database must act in predictable and consistent ways to improve the user's sense of position and control over the system. It is especially critical to help CASHE:HPVS users gain confidence in the system so they are encouraged to consider and retrieve human factors data when designing systems.

Experimental Objectives

This study had the following primary experimental objectives:

1. Determine if users with backtracking tools show a decrease in navigation time over users without backtracking tools.
2. Compare the navigation times for four backtracking tools.

3. Determine if particular backtracking tools were preferred to a standard hierarchical menu more often than other tools.
4. Obtain feedback through a questionnaire on user's perception of the efficiency and ease of use of the backtracking tools.

LITERATURE REVIEW

There is essentially no literature on backtracking through interactive databases. Fortunately, it is a subset of a topic known as user recovery. User recovery changes the state of the system back to a previously existing state to "undo" either user or computer errors. Backtracking is quite similar except that all the necessary states of the system already exist. That is, the computer does not have to calculate or remember the product of the operation to be "undone", only a record of the user's path must be kept.

The first section of this review explores views on the importance of user recovery. The second section provides insight into mental models as they apply to a software user. It outlines the ways in which the gaps between the user and computer may be reduced to decrease the cognitive workload and improve the user's sense of position.

The third section reviews literature on characteristics of user navigation in a database. Disorientation is a common and unsettling problem for database users.

The fourth section draws on the user recovery literature to outline possible algorithms for implementing GO BACK. The user's path can get quite complicated as undos are embedded in undos and the user accesses a component by multiple routes multiple times. If improperly designed, GO BACK could prove to be a source of confusion rather than help! The algorithms were researched to help determine an equitable way to balance simplicity and functionality of the GO BACK feature.

Section five addresses research and positions on the top level interface design issues such as screen design, history list portrayal, and node lists versus graphical representations.

The sixth section concludes the literature review by summarizing research on the level of complexity users want in an undo facility. As a software designer, it is tempting to program for the maximum amount of functionality; in the case of backtracking, this may not prove to be the best philosophy.

The Importance of User Recovery

In any type of human-computer system, it is important to provide the user with a method for reversing the effects of commands and actions. This is especially true with the direct manipulation style of interface. The popularity of direct manipulation is growing mainly because it is often simple to learn, enjoyable to use, and encourages users to freely explore the system (Shneiderman, 1987). Users may learn about the semantics of the system by a hands-on approach rather than reading documentation. "Interacting with the system is often necessary to clarify the user's understanding of it" (Yang, 1988b, p. 457). Direct manipulation provides incremental responses to each user action, and the user can immediately judge the correctness of the results. The incremental response feature would not be as valuable without a function to "undo" the results and backtrack the user to a previous system state.

In many cases, commands have inverses that may be used for manual reversal (e.g. cut and paste). Archer, Conway, and Shneider (1984, p. 16) comment on the inefficiency of manual recovery. "Ad hoc recovery by the user is slow, expensive, and unreliable; automatic recovery by the system can be convenient, reasonably economical, and reliable." Computers are better suited to remembering commands and their sequence. Users should be focused on task issues rather than recovery issues.

Archer et al. (1984) add that errors are not restricted to novice users. The characteristics of the user's errors may change with experience and time but their occurrence does not.

The user may not only need to correct his own errors, but also errors created by the computer, the input devices, or interface. Though system errors are frequently committed by the user, occasional flaws in software and hardware may cause the system to respond in an unpredictable, undesirable manner. For instance, situations may arise when a user employs a type-ahead strategy to quickly make a number of sequential menu selections. Due to system delays, some commands may be inadvertently omitted from the sequence and the user needs a provision to undo *n* commands to reinsert the omitted command.

There is a great deal of agreement that recovery features are necessary in computer interfaces. Leeman (1986, p. 50) says that "there is evidence from the human factors community of the importance of undo." Almost any reference about human-computer interface design mentions the criticality of recovery functions, but they do not explain how best to aid the user or design the feature! The software interface design guidelines by Smith and Mosier (1986) are very comprehensive yet only three recommendations are given about undo: any user action should be reversible, the undo should act on more than just the one most recent action, and the undo itself should be undoable.

Though user recovery is desirable, it should not be overused and may not be applicable in every situation. For instance, undoing an action in a computer-based game or computer-aided instruction package may not always be desirable. Designers may be tempted to incorporate an undo facility merely as a simple solution to compensate for a difficult software problem. In this case, user recovery is provided as a quick fix. A certain number of "bugs" is unavoidable in software design; providing undo capabilities is an easy solution which reduces the criticality of locating and correcting these errors.

User recovery facilities have benefits besides the obvious error correction they provide users. Thimbleby (1990) cites four additional advantages of "undo". First, it aids the user in viewing the interaction as either spatial (a map-like representation of the interactions) or temporal (a linear sequence of events) depending on his or her method of conceptualizing the structure of the system. Second, the semantic differences between the interface designer and the user can be reduced early in the user's learning. Through exploration, interaction, and recovery, the user is more likely to form a more accurate representation of the system. Third, the history of interactions is explicitly recorded; it can be accessed, referenced, and modified by the computer and user. Fourth, as stated previously, users will be more encouraged to explore the system.

Very little research has been done on the actual implementation schemes of recovery support; much of the research remains very general and anecdotal. Yang conducted a survey in 1989 to explore user attitudes and opinions regarding undo and recovery support. It was one of the first attempts at obtaining empirical evidence for the inclusion of recovery features. Yang's questionnaire results confirmed two very general hypotheses: user recovery is important and useful, and the support should be consistent with the rest of the user interface. These findings are not very surprising but the survey at least provided some empirical support for recovery facilities.

In earlier work by Yang (1988b), a taxonomy of user recovery transactions was suggested: retraction, insertion, extraction, replacement, and transposition. Ideally, recovery support should be able to perform each of these five classes of operations. Database navigation backtracking is concerned mainly with retraction, or undoing the n most recent navigation maneuvers. Even though database designers are probably only concerned with this one type of recovery, there are many unanswered questions about how best to implement the retraction.

There are four aspects of the recovery support that demand attention (Yang, 1988b): presentation, internal interface, data structure and algorithm, and recovery technique. Presentation involves the look and feel of the recovery facility from the user's point of view. It includes the way in which the command is issued, the representation of histories, and the system response. The internal interface aspect pertains to the relationship of the recovery code to other components of the system code. The data structure and algorithm deals with formal statements of system response to the various recoveries that may be required. Lastly, the actual recovery technique used to restore a state must be established. With regard to a recovery facility in an interactive database, the major concerns for the interface design are the presentation and the data structure. The interface must be created independently from the system code.

User recovery in a database often means computer-aided navigation assistance. Navigation assistance may be simpler to implement than complex command recovery, but it is no less valuable to the database browser than command recovery is to the programmer. Canter, Powell, Wishart, and Roderick (1986, p. 257) have examined the issue of navigation through databases. They state, "a novice database user is indeed a stranger in a strange land and it is clear that much research is needed in discovering what kind of transport system he needs and which navigational aids are likely to be of most help to him." In an earlier work, Canter (1985, p. 101) commented on how important it is for database users to have a sense of position and "safe routes" throughout a system: "Methods of locating, characterizing, and enhancing the visibility of such landmarks may represent an important future application of the current research." A large number of studies can be found on general database search issues such as boolean operators or document classification schemes. However, it must be emphasized that there is almost no empirical research on less structured database browsing. Therefore, this literature review must draw upon the writings of subject matter experts in areas of cognitive theories, recovery algorithms, and mental models to establish the foundation for this research.

Mental Models of Interfaces

Prior to determining recovery algorithms, it is useful to conceptualize the user's view of an interface. This view is termed a mental model and is defined by Beard and Walker (1990, p. 452) as "an individual's internal set of interconnected ideas that incorporate the understanding of what a system contains, the rules by which a system is governed, and possibly an explanation of a system's behavior." Mental models are developed through experience and interaction. They can significantly affect learning and performance. An example of the power of a mental model is found in using scientific calculators. Some users have adopted a mental model in which they enter formulas in strict algebraic notation: 4+2=. Others have adopted a mental model in which the reverse is true- numbers are entered first followed by operators: 4 *enter* 2 +. This is used in Reverse Polish Notation (RPN). The RPN and algebraic notation models are both so strong that users of one type of calculator may have difficulty using the other. An interface designer's goal is to match system structures and semantics to the user's concept of the system. This is not a simple task considering the large variance in user characteristics. To understand how to equate the user's model and system model, it is helpful to examine Norman's (1986) Theory of Action.

Norman (1986) identifies seven stages of user activity that are necessary to interact with a computer interface:

- Establishing the Goal
- Forming the Intention
- Specifying an Action Sequence
- Executing the Action

- Perceiving the System State
- Interpreting the State
- Evaluating the System State with respect to original Goals and Intentions.

The first four stages involve transforming a thought into a system input. The last three stages represent the steps in determining how the system output relates to the user's original goals and intentions. Each of these portions of the cycle represents a potential for conflict between the user model and the system or designer's model. These gaps are known as the *Gulf of Execution* and the *Gulf of Evaluation* (Hutchins, Hollan, and Norman, 1986 and Norman, 1986). These gulfs are shown in Figure 2. The gulf of execution is bridged "by making the commands and mechanisms of the system match the thoughts and goals of the user as much as possible" while the gulf of evaluation is bridged by "making the output displays present a good conceptual model of the system." (Hutchins et al., 1986, p. 95) Each step in the user activity sequence requires a translation or mapping which must be minimized. Bridges must inevitably be built from the system *and* user sides because of the inherent variability in users' models.

Two components make up the gulfs of execution and evaluation: the semantic distance and articulatory distance (see Figure 2). The first, semantic distance, indicates the disparity between a user's representation of a task and the system's representation. In other words, does a user have to construct a task from available commands or is there one command that acts precisely the way in which the user intended? On the gulf of evaluation, does the display provide the results that the user needs? The other component of the gulfs is

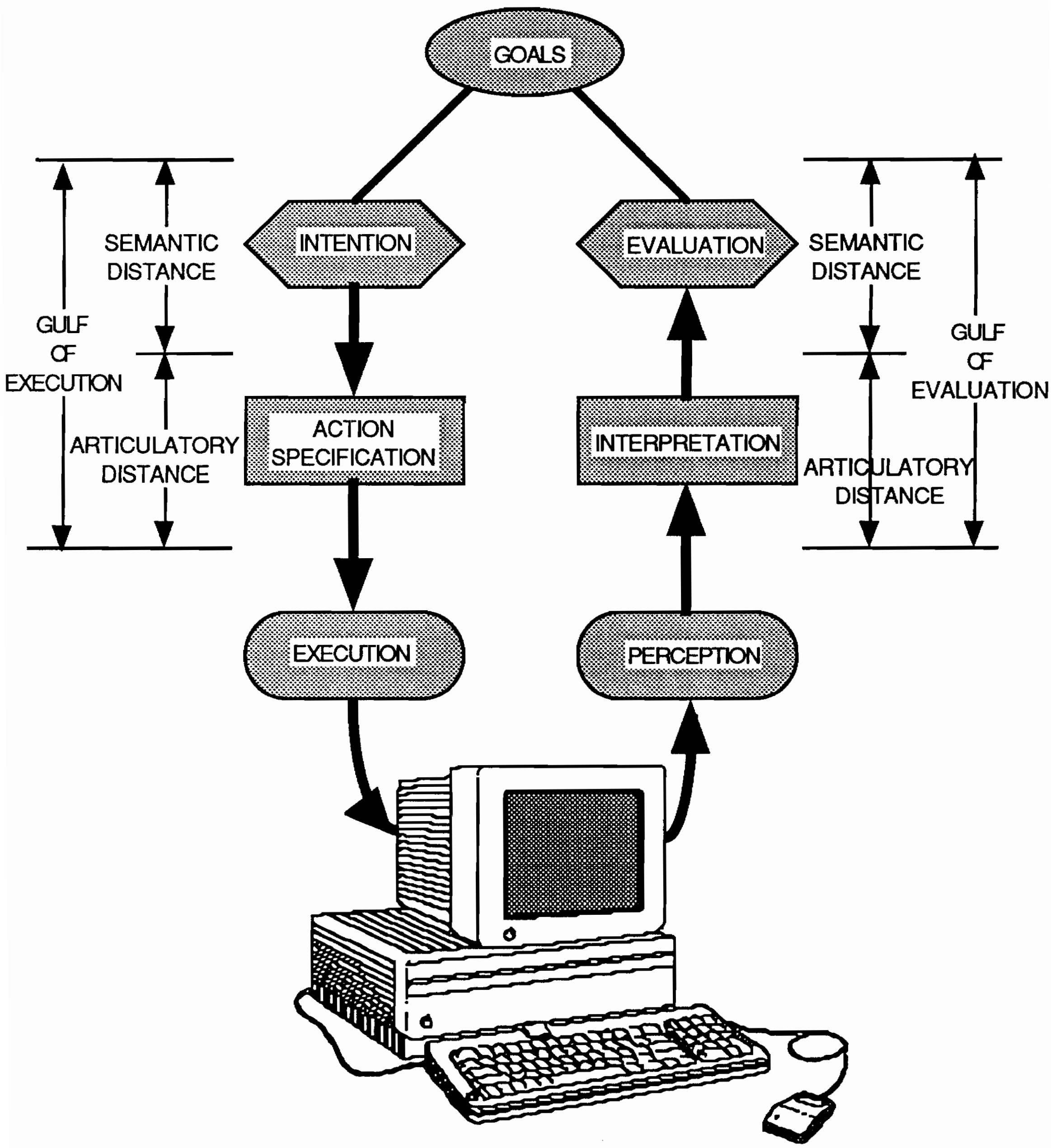


Figure 2. Relationships among distances and gulfs.
(adapted from Hutchins, et al., 1986)

articulatory distance. It refers to the translation of a task into a physical input form or the display of a system response in an appropriate format. In other words, is the form consistent with the meaning? The articulatory and semantic distances are both targets for reducing the magnitude of the gulfs. This metaphor of the gulfs of execution and evaluation provides a meaningful model by which designers can evaluate and improve their interfaces. For the user, experience and interaction with a system will help to reduce the *effects* of the gulfs. However, the *size* of the gulfs of execution and evaluation can only be decreased by altering the user's conception of the system or the designer's representation of the interface. The designer can change the interface semantics or help the user to alter his or her mental model and understanding of the system.

The next issue that deserves attention is how users form and organize their mental models of an interface. Quillian (1968) researched theories on how the human mind creates associations to construct the semantic memory. The semantic memory is described as a series of nodes joined by associative links making up a very complex network. Users rely on this semantic memory to form models of direct manipulation and interactive interfaces. The action-reaction association between user and computer help the user to learn and retain the system principles. Weiser and Shneiderman (1987) note that semantic knowledge is best acquired through this meaningful learning.

Canter (1977) says that people are fairly good at creating "maps in their heads" using the semantic memory. This is fortunate, especially in the context of a database, because people can remember a general picture or structure better than they can remember the hundreds of branches and links. In a map drawing task, Canter found that people drew spatial maps 78% of the time in contrast to the remaining 22% that were sequential in nature. People have a much easier time with spatial relationships and designers have capitalized on this. Perhaps direct manipulation is such a successful interface style because it is easily transformed into spatial relationships of semantic knowledge rather than temporal relationships between syntactic knowledge.

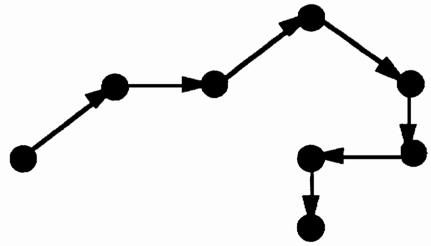
The user's understanding of an interface is critical to maximizing its usefulness (McMath, Tamaru, & Rada, 1989). The designer must be sympathetic to the kinds of biases, stereotypes, and ideas the user brings to a system. The interface should be suitably adapted to meet the needs of the user and not vice versa. Users have powerful semantic memories and create mental models that exert tremendous influence in helping or hindering a user's interaction with a computer system. Understanding the power of the mind is not adequate in itself; users' behavior in computer systems and databases in particular must be accounted for in design.

Searching and Maintaining Position in Databases

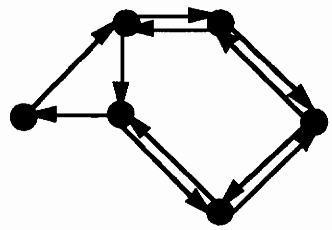
To better understand how to help a user backtrack through a database, it is essential to study and characterize the way in which a user moves forward through the system. Canter, Rivers, and Storrs (1985) devised a taxonomy for classifying the search strategies. They state, "even a casual reading of the data-base literature reveals that data sets are no longer only searched. They are browsed, explored, scanned" (Canter et al., 1985, p. 93). The terminology Canter et al. have selected is useful for describing user navigation: a *path* is described as "any route through the data which does not cross any node twice" (Canter et al., 1985, p. 95); a *ring* is "a route through the data which returns to the node at which it starts- it may include other rings" (Canter et al., 1985, p. 95); a *loop* is a ring not containing other rings; a *spike* is a route upon which the user precisely backtracks through the exact set of nodes in precise reverse order" (Canter et al., 1985, p. 95). These four configurations are shown in Figure 3.

Canter et al. (1985) also devised indices to characterize the navigation of a user: pathiness, ringiness, loopiness, spikiness, ratio of different nodes visited to total number of node visits, and ratio of number of nodes visited to number of nodes in the system. The most interesting part of Canter et al.'s research was their hypothesis that a high "spikiness-quotient" (SQ) may be associated with a fear of getting lost in the system structure. They were also able to associate a high or low SQ with the five classes of search strategies: scanning, browsing, searching, exploring, and wandering. Scanning (users looking at variety of information but not in depth) and searching (seeking a particular target) were two strategies with possibly high SQs. Low SQs were associated with exploring (examining extent of database), browsing (untargeted), and wandering (unstructured) strategies. The relationship between the indices and types of user navigation are only speculations. The more useful portion of the literature is the descriptive terminology and compact taxonomy for categorizing navigation strategies.

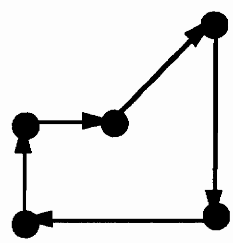
While it is important to understand the mechanics of user navigation, the navigation itself is a secondary or support task. Beard and Walker (1990) note that the structures for navigation should be designed to use as little



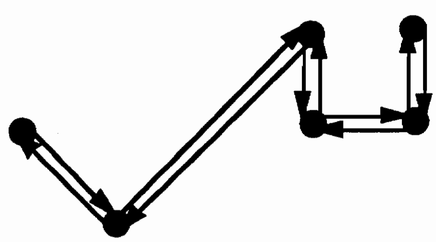
a. A path



b. A ring



c. A loop



d. A spike

Figure 3. User navigation classifications.
(adapted from Canter et al., 1985)

"cognitive overhead" as possible. Unfortunately, the anxiety from interacting with a powerful computer system often causes a user to focus on navigating in the system and not the task at hand. Losing one's sense of position is a real concern with many database users, especially novices.

Human nature dictates that people seek an understanding of all types of occurrences in the world. Williges, Williges, and Elkerton (1987, p. 1421) say that "one fundamental aspect of human behavior is that humans seek structure or organization in their environment, even in cases where none exists." They stress that this element of structure is an important feature in software interfaces. Other researchers support this belief especially in the domain of databases. Field and Apperley (1990, p. 135) discuss database searching and say, "this sequential process gives use to two of the fundamental questions in database navigation, 'where am I?' and 'How do I get back to x?" Users are constantly struggling to maintain a sense of position. As stated previously, a concern for the mechanics of navigation is an additional, unnecessary cognitive workload for the user.

User-centered interface design is a philosophy in which the user is given ultimate control (or feeling of control) over the system. Fitter (1979) says for users to feel this control, they must understand the interface, know where they have been, where they are, and where they can go from where they are. This conceptual knowledge of the interface and ability to navigate can be acquired over time, even with the most ill-designed interface. However, "it should also be considered that in the case of the naive user, the means of traveling may well be obscure as well as the destination" (Canter et al., 1986, p. 249).

Novice users may need help in forming their mental models and mapping them to the system from the moment they sit down at the computer.

User Recovery Models

There has been little or no published research on backtracking and browsing in an interactive database. However, it is worthwhile to examine formal undo structures as they exist in other environments. The parallel was drawn earlier- backtracking in the graphical database is just a specific case of undo in which all tasks to be undone are navigation tasks. The first section deals very briefly with system-level techniques for executing recovery and subsequent sections deal exclusively with the interface-level strategies.

General Recovery Strategies. The earliest form of computer recovery (Leeman, 1986) was setting checkpoints in a procedure for solving a differential equation on an ENIAC computer. A checkpoint is essentially a copy of the contents and state of the system at a given point in time. One extreme of checkpoints is the complete rerun strategy in which the whole history of commands and interactions are resubmitted by the computer. Archer et al. (1984, p. 9) states "the significance of the complete rerun strategy is that it establishes that recovery is possible. Only a concern for improved performance motivates consideration of other strategies." Obviously, this full-rerun approach is inappropriate if the user merely wanted to perform a simple undo such as recalling a section of deleted text. The other extreme of the checkpoint strategy is setting a checkpoint after each command. A "clean" copy of the state of the system would be stored after each command so no commands need to be reissued to restore the system. Undoing would merely restore the previous checkpoint. This scheme could be modified so some checkpoints are discarded as new ones are formed to reduce the memory requirement. The checkpoint strategy could be further modified to include a record of user transactions between checkpoints. This is typically called a transaction log or history record (Yang, 1988b). This will be discussed in greater detail later on.

Full checkpoint strategies are generally inefficient in highly interactive interfaces; the computer and user are constantly interacting to mainly cause changes in only small portions of the system. A partial checkpoint strategy is much more suitable; it involves storing only the portions of the system that were modified. It is more efficient than full checkpointing.

The last method of recovery is issuing inverse commands. From a system's perspective, this method is most efficient because it does not require the sizable allocation of memory for storing prior system states. However, if a user issues the commands, he or she must do so in precisely reverse order. Unfortunately, the human memory is far from reliable and should not be given the duty of remembering commands and their order. In the case of complex applications, many commands may not have simple inverses. Computer support is undoubtedly necessary. The following section deals with recovery strategies at a higher level with which the user has direct interaction.

Recovery Terminology. The previous discussion dealt with some methods of recovery from a general system viewpoint. The next logical step is to examine recovery support from a more detailed, procedural perspective. First, some terminology will be introduced to reduce confusion. The user issues two types of commands. A *task oriented* command implies "the state of a working object is changed to a new state which approaches the task-desired state," (Yang, 1988b, p. 459) whereas a *recovery* command changes the state of an object to a previous state. A *history* is an unaltered record of every command (task and recovery) that the user has submitted. Depending on the particular recovery feature, the *script* may be either a log of all effectual task commands (Archer et al., 1984) as is the case for simple undo truncate models (described below) or the script is composed of all effectual task commands and all task commands that are available via undo/undo or undo/redo (also described later). The *state* of the system is then the result of the script and different scripts may lead to the same state. These explanations will become more clear as examples are presented in the following sections.

When a user employs recovery support in an interface, he or she is altering the script of the system. Designers affect what control users have over the script. The next section describes the types of modifications that are possible with different recovery models.

Script Modification. The choice of the recovery mechanism dictates how much flexibility users have in modifying a script. The most restrictive type would only allow the user to add to the script; there is no provision for recovery. The least restrictive would be complete recovery support that provides the user complete control over script modification. Presently, due to constraints within a computer system, there is a limit to the manipulating a user can do with a script. Some commands are restricted in their position to precede or follow particular commands or sequences of commands. For instance, a user cannot very well delete a CUT or COPY command that immediately precedes the first PASTE command. Without performing a CUT or COPY, PASTE is not even available as an alternative because nothing resides in the buffer! Obviously, this totally unrestricted recovery is not a viable solution.

Thimbleby (1990) outlines four general options in script modification that are much more feasible from both the user and computer points of view. The first is *block-truncate*. This corresponds to a periodic full checkpoint strategy. The user is permitted to undo an entire sequence or block (not just portions of it) of commands that were issued after the prior checkpoint. Users manually perform a block truncate when they save back-up versions of a document. After altering the contents of the current file, they may abort an editing session and revert back to the most recent back-up.

A *single truncate* methodology allows the user to undo the last command from the script. Only the effects of task-oriented commands are removed from the script. In other words, an undo cannot undo itself. Once a command is undone, it must be redone manually; there is no redo feature here. *Truncate** (the asterisk indicates that truncate can be applied to zero or more commands) allows the user to undo a specified number of commands provided the number of commands in the script is not exceeded. With single truncate and truncate*, once a task command is reversed, it cannot be reinstated via the undo facility.

Undone commands can be redone via the script modification policy of *truncate/redo*. This is a *truncate** model in which the user may re-issue the truncated commands via the redo function. In this manner, the memory requirement of the user is significantly reduced as the computer "remembers" the commands and their order. It is a useful feature, but quickly becomes complicated as many questions arise. What happens if the user undoes two commands, inserts a new command, and tries to "redo" the undone commands after an insertion? These questions will be addressed to some degree in the following two subsections.

All of these models of script modification are complicated by the fact that not all commands can readily be undone nor do they all need to be. Not only must the user be aware of which commands may be undone, but he or she must understand the mechanics of the modification policy. The two general classifications of script modification are linear and tree-based undo structures.

Linear Recovery Mechanisms. Truncate/redo script modification offers users a simple, linear approach to error recovery that helps to minimize the user's memory load. There are two types of truncate/redo: history undo/undo and linear undo/redo.

History undo/undo is a single truncate model in which the undo command reverses the effect of the most recent command regardless of whether it is a task command or recovery command. Thimbleby (1990) calls this model the "flip-undo" and notes it is simple to implement, easy for users to understand, and consistent. It is based on one rule: it undoes everything including itself. An example of *history undo/undo* is shown below.

A user issues commands C1 and C2 followed by an undo. The undo reverses the effect of C2 leaving the script with just command C1 being "active". The notation adopted by researchers is that any effectual task commands are included in the script as well as "undone" task commands that could possibly be reinstated by an "undo" (or a redo depending on the recovery scheme). The last effectual task command is marked; here it is shown in boldface type.

history: C1-C2-undo
script: **C1**->C2
state: C1

Next, the user issues command C3 followed by undo and undo. The first undo reverses the effect of C3 while the second undo reverses the effect of the first undo thereby reinstating C3 as follows:

history: C1-C2-undo-C3-undo-undo
script: C1-> **C3**
state: C1-C3

Notice that the command C2 is removed from the script because it has not affected the system state nor is it available via undo. The undo is limited to the single most recent command. It is quite simple but Thimbleby notes that this clean simplicity is marred by the fact that designers often only apply undo to "major" commands. The user is left to discover the domain of the particular undo.

Some history undo/undo systems prompt the user for the number of commands to undo. This may enhance the history undo/undo functionality but it must provide the user with a visual history list or rely on the user to remember the number of commands that must be undone! Says Vitter (1984, p. 40), "the power of history undo/undo is that UNDOs can return the state of the system to any former value. The severe limitation is that UNDO can only return the state to some former value." Issuing undo once will remove the effect of the last *n* commands issued. Issuing undo immediately subsequent to the first will undo the undo and revert the system to its original state. The history undo/undo model is useful for what Vitter terms a "change of heart" in which the user begins to modify the script then changes his or her mind and wants to revert to the state that existed before the script was modified. This model is poor for inserting commands in the midst of a script. Once the *n* commands are undone and the chain is broken with a command entry, the undone commands cannot be redone automatically. Hence, the history undo/undo is simple to implement but very limited in its functionality.

A more useful variety of truncate/redo is the *linear undo/redo*. In this type of recovery, undo and redo are called meta-commands (Vitter, 1984) and only operate on task-oriented commands, not recovery commands as in history undo/undo. An example is provided below. The user issues four commands C1, C2, C3, and C4 and realizes that C5 should have been issued between C2 and C3. One undo will reverse C4 and another undo issued immediately will reverse C3.

history:	C1-C2-C3-C4-undo-undo
script:	C1->**C2**->C3->C4
state:	C1-C2

The user may then issue C5 and reinstate C3 with a redo and C4 with another redo.

history:	C1-C2-C3-C4-undo-undo-C5-redo-redo
script:	C1->C2->C5->C3->**C4**
state:	C1-C2-C5-C3-C4

Linear undo/redo is best suited to simple truncations or command insertions but it cannot fix a "change of heart". Its major advantage is that it allows the user access to as much of the script as the computer has stored. History undo/undo normally only allows the user to modify the last command issued. With linear undo/redo, the user may undo *n* commands, insert commands, then redo *m* previously issued commands. Redo acts on a last-in-first-out principle: the most recent undone command is redone first. Linear undo/redo is much more flexible than history undo/undo but the semantics can be more difficult for the user to understand.

Tree Recovery Mechanisms. Both the history undo/undo and linear undo/redo rely on a linear representation of the interaction between the user and the system. Vitter (1984, p. 39) says these "former undo/redo packages organized recovery information in a linear way that often did not reflect the user's intentions." To compensate for this shortcoming, some designers tried to employ a tree-like structure to provide more flexibility to users. Toriya, Satoh, Ueda, and Chiyokura (1986) used a graphical tree to aid users in obtaining their desired recovery in solid graphical modeling operations. A more general tree-based recovery system was developed by Vitter (1984). He devised a methodology called US&R- Undo, Skip and Redo to compensate for the major weakness of linear models. US&R is suitable for command insertion, change of heart, command substitution, selective undo, command rearrangement, and selective rearrangement. As in linear undo/redo, the undo, skip, and redo are meta-commands; they can only operate on task commands and not on each other. US&R is extremely flexible though it does require some decision-making on the part of the user in some instances. It maintains a graphical log of the complete history of the system including branches that have no net effect on the system.

US&R is best understood by example. Assume that a user has executed a sequence of commands, C1-C2-C3-C4. Examine the way in which the US&R commands modify a script and state. Again, the last effective command submitted by a user is indicated by bold type.

history: C1-C2-C3-C4
script: C1->C2->C3->**C4**
state: C1-C2-C3-C4

To begin, the user "undoes" C2, C3, and C4 by issuing the UNDO command three times.

history: C1-C2-C3-C4-undo-undo-undo
script: **C1**->C2->C3->C4
state: C1

The system is back to a state in which only C1 has been executed. Next, the user enters commands B1 and B2.

history: C1-C2-C3-C4-undo-undo-undo-B1-B2
script:
```
        C2->C3->C4
C1---{
        B1->B2
```
(last effective command: **B2**)
state: C1-B1-B2

The system records the B1-B2 branch off of the C1 node. It maintains both branches for the user to access. Now the user enters "undo" twice - the first undoes B2, the second undoes B1.

history: C1-C2-C3-C4-undo-undo-undo-B1-B2-undo-undo
script:
```
        C2->C3->C4
C1---{
        B1->B2
```
(last effective command: **C1**)
state: C1

Now the user selects "redo." He or she is prompted for the choice of branch to reinstate. After selecting the C2-C3-C4 branch, the user can "skip" C3 (as indicated by the strike-through font) and "redo" C4 for the following result:

history:	C1-C2-C3-C4-undo-undo-undo-B1-B2-undo-undo-redo-?-skip-redo
script:	C1---{ C2->~~C3~~->**C4** / B1->B2
state:	C1-C2-C4

The US&R model is designed to accommodate the non-linearity of human decision making and intentions. Vitter (1984) notes that a primary goal of his research was to show the suitability of a branching model over a linear model for supporting user recovery. The technique is an admirable undertaking, but it is not without its flaws.

The increased functionality and flexibility of US&R is achieved at the expense of complexity. Thimbleby (1990) points out several weaknesses in the US&R. He notes that the representation of the script can be difficult for a user to interpret. Often times, it is easier for a user to determine how to manually manipulate the actual state rather than the script. Also, the users are given a choice of script branches emanating from the current node but they cannot be expected to remember the precise consequences of taking each branch. The interface for US&R recovery support would have to be quite expertly crafted for users to invoke it. The semantics of the undo feature must be clear to the user. One way to reduce the semantics problem is to provide recovery explicitly instead of implicitly. This can be achieved by giving the user access to the history list so he or she can directly manipulate it. The following discussion is devoted to the topic of history lists.

Interface Design Issues in User Recovery

History List Interface. History lists, transaction records, and command logs are all terms used for a computer-maintained listing of executed commands presented in temporal order. Field and Apperley (1990, p. 135-136) studied navigation with hierarchical menus and have found that "if the selection history is shown to the user, it is suggested that users have a much better contextual picture of their current position within the database, and that menu navigation ceases to be a short-term memory exercise." In each level of the menu hierarchy, a full list was maintained of all the menus that had been used to navigate to the current menu. The user could "deselect" a menu item to backtrack up the menu tree. Field and Apperley found that this explicit selection log gave subjects a significant advantage in performing benchmark search tasks over subjects performing without the log.

There is very little research exploring the willingness and ability of users to manipulate a history list. Greenberg and Witten (1988) examined how often people access the UNIX history list called the *csh command interpreter*. The most interesting result was the frequency with which different classes of UNIX users consulted the history list: 20% for novices, 36% for non-programmers, 71% for computer scientists, and 92% for experienced programmers. Although UNIX is not the most representative platform for usability studies, the study does show some type of relationship between a user's experience and frequency with which they consult the history list. However, the actual design of the UNIX command interpreter interface may have greatly affected the usability of the history for inexperienced users. The presentation of this history list is critical to user acceptance.

Yang (1989) surveyed university employees about their perceptions and opinions about recovery and command reuse support. The most revealing portion of the survey was the responses to the open-ended questions. The respondents were asked how they would like history information displayed. Some suggested a state transition network or an ordered list in a window. The window could be permanent, called-up, or a pop-up window. The list could display commands or icon names and the user could set the number of commands that were visible at a time. Other respondents recommended a natural language description with graphical simulation. Lastly, some merely indicated that only the last executed command need be shown. The recommendations for displaying the undone commands included somehow highlighting the undone commands or maintaining a separate list of undone commands. No information is provided about the frequency of certain responses but it is interesting that overall the comments stressed ordered lists rather than graphical or tree-like representations. Also, the responses show that there are no hard-and-fast stereotypes regarding how users expect to see this information.

Another pertinent issue in displaying histories is whether displays should be static or dynamic. In an ordered list, should the current command be displayed at the top of the list or in its position relative to all issued task commands (even undone commands). Furthermore, should duplicate commands be pruned off the list and if so how? One method saves the command in its most recent position and one shows the command in its original position (Greenberg and Witten, 1988).

Undo Interface. There is, once again, little research or guidance on recommended procedures for invoking recovery commands. General human-computer interface principles certainly apply - such as recovery features should match the style of the rest of the interface. Yang's (1989) survey data on recovery features provides insight into what users prefer for issuing undo commands. The recommendations about the physical form of an undo function include a pop-up menu, dialogue box, icon, or function key. Some respondents recommended a confirmation before undoing while others suggested a direct undo by picking a point in a history list.

Range of Undo. Recovery features come at the expense of extra programming time, additional computer memory requirements, increased overall system complexity, and longer learning times. Therefore, it is not necessarily optimal to construct the most comprehensive and functional recovery system. There is no need to provide a history list that is one hundred items long if users are never comfortable enough to rely on recovery support for commands issued so far in the past. It is generally accepted that the human's working memory can retain about seven, plus or minus two "chunks." (Miller, 1956) One could hypothesize that an interactive system user could remember the last 7 ± 2 commands issued, if no interference were introduced. Greenberg and Witten (1988), in their study of UNIX history list use, found that not including programming novices, users referred only to the last five commands in 79-86% of the time. In most instances, they really only looked to the two most recently issued commands. When the computer "undoes" more than a few commands, the user may become disoriented and lose proper perspective for the state of the system. For large recoveries, the user may prefer to abort and retrieve an old checkpoint or back-up or merely resurrect the system manually. This leads to the last concern pertaining to the tradeoffs involved in determining the level of complexity that should be selected for the recovery facilities.

Functionality versus Simplicity

Simple recovery models such as history undo/undo and linear undo/redo provide simple, predictable, albeit restricted capabilities. They don't do a great deal, but they don't require much cognitive exertion on the part of the user. On the other hand, there are highly functional models such as Vitter's (1984) US&R model that have fairly complicated semantics that may be difficult for users to learn. Yang (1988b, p. 468) comments that "this raises the open question as to what the right trade-off should be between having a powerful undo support facility and having a simple interface realization of it." In his later research, Yang (1989, p. 446) wrote, "people's belief whether it works or not, their knowledge of how to use it, and their attitude to whether the complexity or even risk of using it is worth the effort of bringing it to bear, affect their use of it. It is the users themselves who determine what is usable and what is not." Usability testing is necessary to ensure that an interface is clear and predictable. Designers must defy the temptation to rely on their own judgment about user preferences and expectations.

Yang (1989) set out to investigate whether experienced users preferred the more complicated, flexible recovery and novices preferred simpler, more structured models. His survey showed that in general, all classes of users tended to rate the simpler recovery support as more useful. Unfortunately, survey data alone is not completely reliable. In this case respondents rated recovery systems based on written descriptions of them rather than actual interactive experience.

Thimbleby (1990) notes that users may not risk using even a simple system if it is inconsistent or unpredictable. Thimbleby (1990, p. 264) says, "the aim now is to make undo simple and reliable in use; perhaps it will be slow but the aim is to ensure that the user may undo everything with well-placed confidence." Yang (1989) predicts that the functionality/simplicity tradeoff is application and user-dependent. It is perhaps useful to conclude this section by citing the interface requirements for undo as defined by Thimbleby (1990). The user interface should be simple, general, and unrestricted. Any limitations should be clearly obvious to the user. Lastly, any recovery facility is useful; even a bad system can be made more usable with some undo features.

Discussion

User recovery is an inarguably necessary component in almost any computer application from text editors to statistical packages. Reference databases are probably no exception. Users of a database such as CASHE:HPVS primarily scan and search for information which is characterized by a high spikiness quotient (Canter et al., 1985); this means that users have a fear of getting lost and frequently retrace their path through nodes manually. This finding may indicate a potential need for computer assisted backtracking

Many approaches have been developed to aid user recovery in a variety of environments. These techniques range from singular command truncations to complex tree-like representations from which the user may customize a recovery. User recovery in a computerized database is marked by one significant difference: unlike many other interface environments, users will always have an option to navigate manually to backtrack. For instance, a TABLE OF CONTENTS, index, or key word search may be employed if the user remembers the particular characteristics of the entry.

The computer can provide backtracking support, but it is important to re-emphasize that this capability is not required for system integrity. Therefore, the backtracking mechanism must have an adequate perceived worth for people to be willing to invest the time to learn to use it. The value can be enhanced by providing the correct balance of simplicity and functionality. Systems such as Vitter's (1984) US&R or Toriya et al.'s (1986) graphical tree recovery aids are highly functional but also could prove too costly in terms of memory and programming requirements to be worthwhile in a database application. History undo/undo models are very simple to use but quite limited in functionality.

For this study, a linear undo/redo type of framework offered a suitable balance of simplicity and functionality for users in a hierarchical database. As there has been little research in this area, it seemed best to investigate whether simple backtracking support offers performance advantages or is preferred over other navigation techniques first. Subsequent studies can be used to investigate effects of more complicated backtracking tools.

There are a number of facts supporting the decision to chose linear undo/redo rather than undo/undo or tree-based structures. First, the idea behind CASHE:HPVS is to simplify access to human engineering information. The interface must be comprehensible for all groups of users including novices and intermittent users; this precludes overly complex recovery support such as tree structures. Second, an undo/undo strategy would limit the backtracking to acting as a toggle between the present node and the previous node. Third, the consequences of user error in database navigation are minimal. The functionality advantage of undo/redo probably outweighs the slight increase in complexity. Based on these facts, the general linear undo/redo model was selected to regulate the backtracking.

Two unresolved issues still remain and were the source of this study. The first issue is whether the navigation history should be presented to the user explicitly or implicitly. That is, should the user be able to view the navigation script (explicit history) or are the computer responses sufficient to describe the action (implicit history)? With an implicit history list, the computer would follow a backtracking path that was precisely the reverse of the path taken by the user beginning with the present node. The user would choose the stopping point but could do nothing to alter the path taken. With an explicit history, the navigation history is presented as a sequential, chronological node listing. The user may select a specific destination from the history list thus by-passing the intermittent nodes. The implicit history presents a much more streamlined front-end and may be preferred by users with good memory skills or mental models of the database. The explicit history may require a smaller cognitive workload but it could also be cumbersome to read and manipulate. The question of an implicit versus explicit navigation history has not been addressed by formal research in the context of a hierarchical database and was selected as a factor in this research.

The second important research issue is whether the backtracking should take place at the lowest level of the database, the component, or its parent level, the entry. Though GO BACK could be used to traverse across, up, and down the hierarchical database, it is most functional to confine backtracking to either the entry or component level. The database is very flat; from an index or TABLE OF CONTENTS, the entry may be reached and from there any component may be accessed. From any component, the TABLE OF CONTENTS or index may be reached with the press of a button. The user only needs support to reach previously viewed components or entries but should not need help finding the TABLE OF CONTENTS. Therefore, the user's navigation history need not be littered by records of visits to the various indices. Smith and Mosier (1986) argue that any action should be reversible which could include, for instance, aborting a trip to a mistakenly visited TABLE OF CONTENTS. This type of commission error will be handled through a cancel button rather than GO BACK. Therefore, for this study, backtracking will be limited to entry and component screens; is the user more efficient or comfortable backtracking at the lowest, or component, level or the parent, or entry, level? This is the second major variable in the study.

METHOD

Subjects

Forty adult subjects (18 female, 22 male) participated in this study. Each subject was required for one session that lasted no longer than two hours. Subjects were between the ages of 18 and 32 and were free from physical impairments which could have inhibited their ability to manipulate a mouse or view a computer monitor. Subjects were not screened for any specified amount of computer experience.

Apparatus

A fully-operational database was developed using HyperCard™ 2.0. It was run on a Macintosh™ II computer with a standard one-button mouse and a 21" monochrome monitor. The database contained hypothetical information about imaginary states and was designed to model the CASHE:HPVS interface as closely as possible. The next two sections discuss the specifications of the experimental database in more detail.

Database Design. A sample screen of the experimental database is given in Figure 4. The left-hand portion is the database display which mimics the CASHE:HPVS interface. The right-hand portion is the task window displaying questions and the multiple-choice answers. As with CASHE:HPVS, entries are titled at the top of display and the "Brass Plate" section is found in

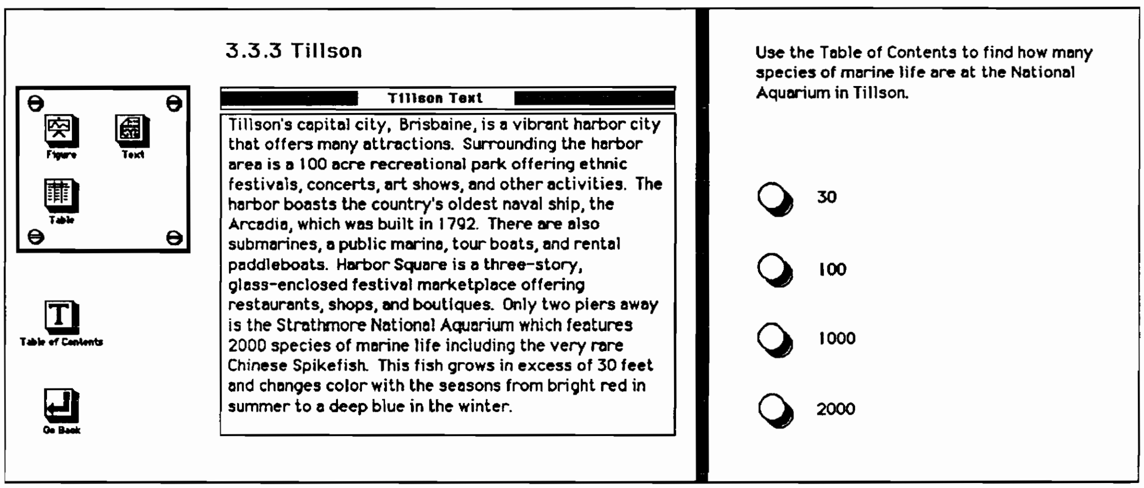


Figure 4. Example text component

the upper left-hand corner; the component icons are identical to the icons used in CASHE:HPVS. Similarly, the GO BACK icon in the experimental database is identical in appearance and placement to the one used in CASHE:HPVS. All buttons automatically hilite with inverse video when the mouse is depressed in the active region of the button.

The database was developed in the context of a travel and vacation database in the hope of promoting active engagement of subjects. A potential user could browse through it to find possible vacation spots, identify things to do, find out about the weather, or learn about the flora and fauna. The database included the states of three countries: Macaye, Kinsela, and Strathmore. Each of the countries was made up of four provinces which were in turn composed of four states. Hence, there were 3x4x4 states totaling 48 states. The state names were each single words of two to three syllables. The hierarchy of countries, provinces, and states is shown in Table 1.

Each entry or state had three components of information: text, table, and a figure for a total of 144 components. Figures 4, 5, and 6 show sample text, table, and figure components from the imaginary state of Tillson. The text component was automatically displayed when an entry was accessed, much as is done in the CASHE:HPVS system. The database information was organized in a hierarchy as shown in Figure 7.

Information within each component type was displayed in a similar manner for each state. Approximately the same amount of information was given in each component. Text was shown in a twelve point Geneva font to simulate the CASHE:HPVS environment and promote readability. To prevent

TABLE 1

Hierarchy of Countries, Province, and States in Experimental Database

Country	Province	State			
1. Macaye	1.1 Potter	1.1.1	Thorson	1.1.3	Harkness
		1.1.2	Klipper	1.1.4	Flint
	1.2 Quenton	1.2.1	Brittany	1.2.3	Bennett
		1.2.2	Sandstrom	1.2.4	Fairchance
	1.3 Atkins	1.3.1	Grafton	1.3.3	Watkins
		1.3.2	Foster	1.3.4	Churchill
	1.4 Hancock	1.4.1	Amhearst	1.4.3	Sutton
		1.4.2	Griffith	1.4.4	Theodore
2. Kinsela	2.1 Randolph	2.1.1	Woodward	2.1.3	Lockwood
		2.1.2	Canterbury	2.1.4	Sassinger
	2.2 Brewster	2.2.1	Athens	2.2.3	Hawthorne
		2.2.2	Gilmore	2.2.4	Ironto
	2.3 Oakmoor	2.3.1	Fairmont	2.3.3	Appleton
		2.3.2	Nottingham	2.3.4	Deerfield
	2.4 Taper	2.4.1	Zimmer	2.4.3	Vaughn
		2.4.2	Partridge	2.4.4	Fishwick
3. Strathmore	3.1 Galloway	3.1.1	Kaufmann	3.1.3	Armstrong
		3.1.2	Radford	3.1.4	Bluefield
	3.2 Wycliffe	3.2.1	Westwood	3.2.3	Northrup
		3.2.2	Rawlins	3.2.4	Costello
	3.3 Enderby	3.3.1	Pamplin	3.3.3	Tillson
		3.3.2	Mitchell	3.3.4	Greenwalt
	3.4 Pembroke	3.4.1	Yorktown	3.4.3	Shepherd
		3.4.2	Ridgewood	3.4.4	Braxton

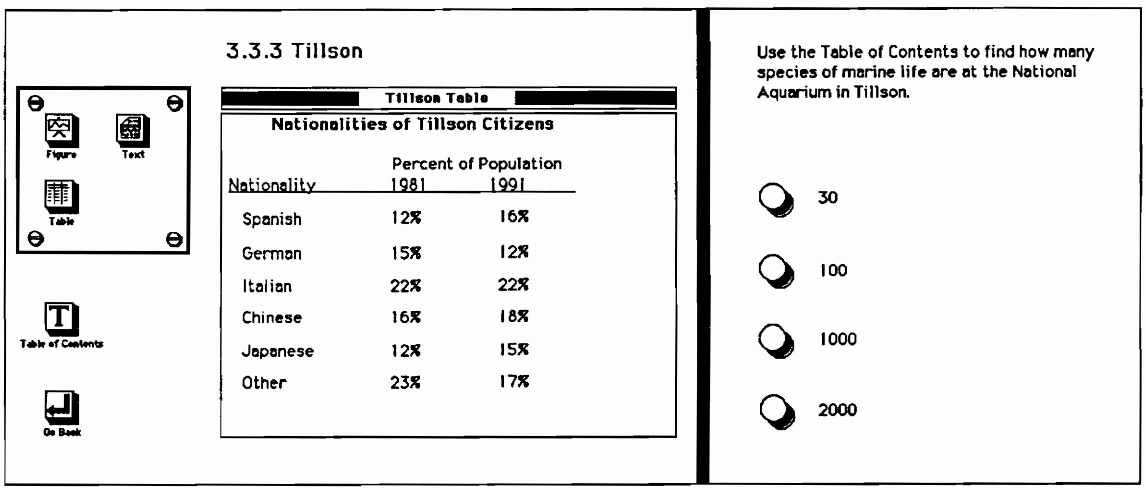


Figure 5. Example table component.

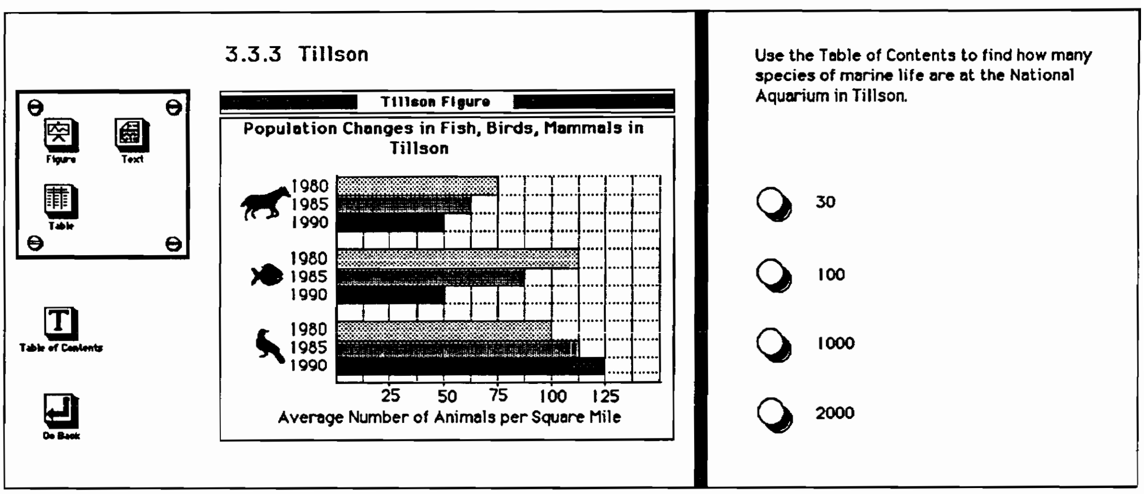


Figure 6. Example figure component.

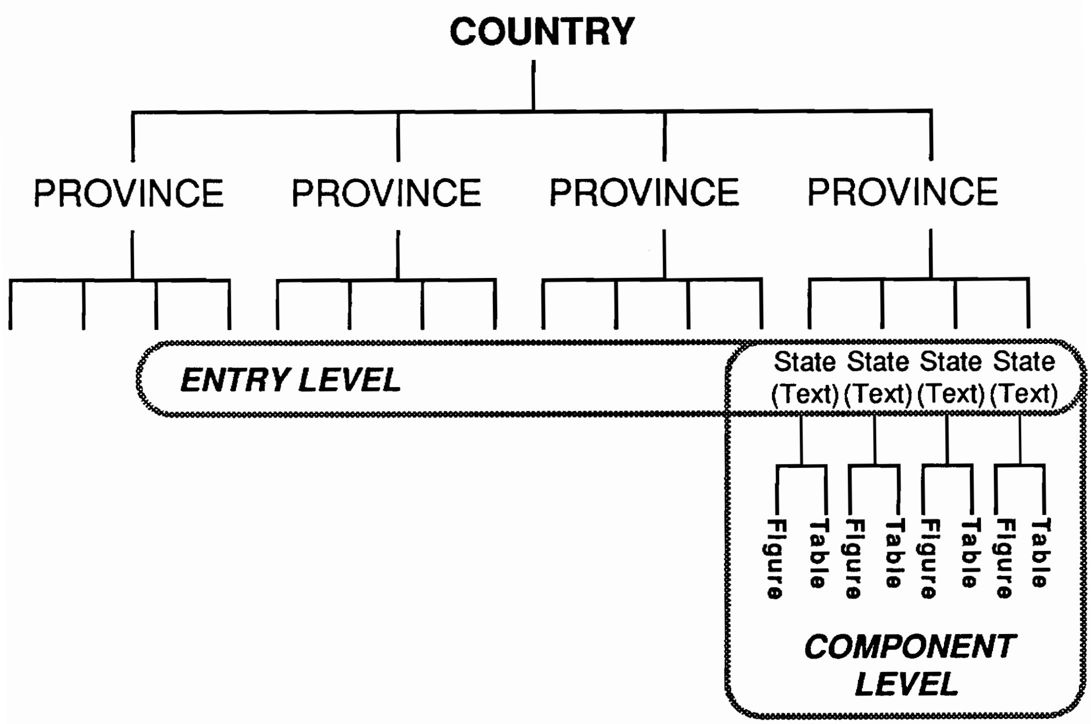


Figure 7. Experimental database hierarchy.

subjects from blindly searching among components, some guidelines were developed for possible information location.

The text component gave information about tourist attractions and events to see or do. The text components may have suggested restaurants, night spots, music events, festivals, or sporting events. The tables gave information about the state and its people: population statistics, historical events, geographical facts, or educational trends. The remaining component, figures, had items such as weather patterns, wildlife populations, vegetation, and topographical maps. This small degree of predictability was included because information in CASHE:HPVS is distributed between components in a similarly predictable manner. The emphasis of the study was on navigation and the information extraction task was only used to give the task credibility.

Database Navigation. Navigation through the database was permitted only by the TABLE OF CONTENTS or the GO BACK button(s). Only one database window was visible at a time: TABLE OF CONTENTS, history list (if applicable), or a single component. This prevented users from merely reactivating windows to navigate. The TABLE OF CONTENTS button and GO BACK buttons were present on all components.

The state entries were all accessible via a hierarchical TABLE OF CONTENTS. A sample selection of the TABLE OF CONTENTS is shown in Figure 8. The TABLE OF CONTENTS was directly available from any component in the database. Users could browse through the hierarchical TABLE OF CONTENTS by single-clicking a country to view its provinces and single-clicking on a province to view its states. As each province or country was selected, it was

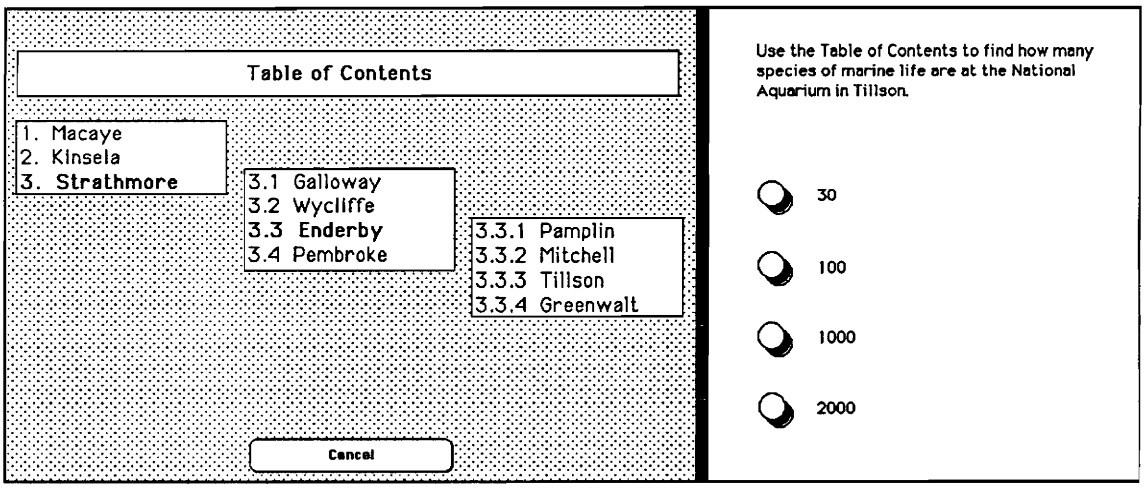


Figure 8. Example database menu hierarchy

marked by boldface type and its respective components appeared in an adjacent rectangle. The user could freely browse throughout the TABLE OF CONTENTS to locate a particular state before committing to view the entry by single-clicking on the state itself. The user could only view the provinces of one country and the states of one province at a time. Subsequent selections resulted in the disappearance of other selection trails, not unlike what is done in CASHE:HPVS.

Four distinctive, but functionally equivalent, backtracking tools were developed for this study. They are: *component-history list, component-no history list, entry-history list,* and *entry-no history list.*

The first two backtracking tools, *component-history list* and *entry-history list,* allowed the subjects to view their history list and select an entry directly. The first of these conditions, *component-history list*, worked in the following way: when the subject pressed GO BACK from a component screen, a scrollable history list was displayed that contained an ordered listing of all the components he or she had accessed. When the subject viewed the history list, it appeared scrolled to the top so the most recent portion of the chronology was visible. This component level history list is shown in Figure 9. The subject then clicked once on the line that contained the desired component. That component was immediately brought into the database viewing area.

The second backtracking tool was *entry-history list*. This condition was also characterized by a single GO BACK button that accessed a history list; however this list contained only entry titles (state names) and not individual components (see Figure 10). From the list, subjects could click on a line to reach the default (text) component of the entry. If their destination was the

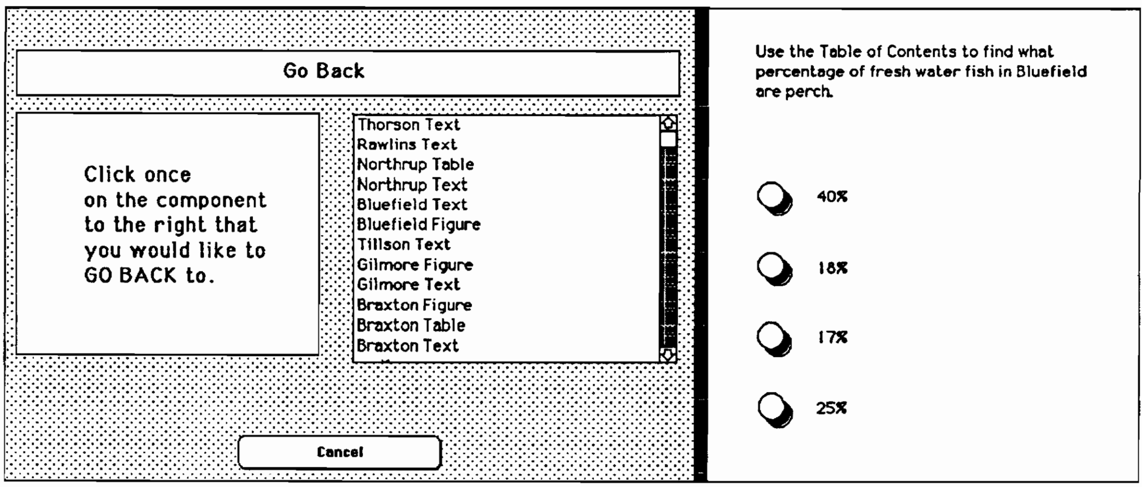


Figure 9. Component history list

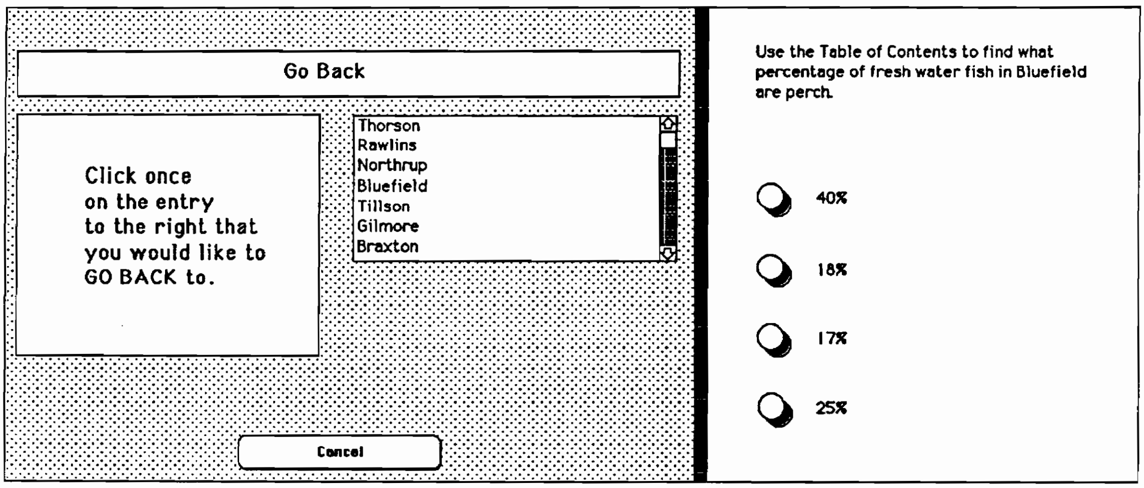


Figure 10. Entry history list

table or figure, they had to select the component via the brass plate control section.

The potential for choosing the incorrect component might perhaps be greater for the *entry-history list* group than for the *component- history list* group. Members of the former group might have difficulty remembering which component they had viewed that contained the needed information; in the component group, perhaps the choice is automatically narrowed down for the subject when only two components of the three appear on the history list. The probability of choosing the correct component is larger when only two components are possibilities rather than three.

There were never any adjacent duplicates of entry or component names in the history list. For instance, for the entry level, even if the subject went from Thorson--text to Thorson--figure to Thorson--table, the Thorson listing would occur only once, not three times in a row in the entry history list condition. Similarly, if the subject repeatedly clicked on Thorson's table icon (mistakenly), the component would only appear once on the history list. It seems likely that this approach is representative of those used in many databases and it certainly reduced unneeded information from appearing on the history list.

The remaining two tools, *component-no history list* and *entry-no history list,* required that the user be provided with a GO BACK button as well as a GO FORWARD button. This was to ensure that both the history and no history groups had the same functionality available to them. Without the GO FORWARD button, users may have accidentally by-passed the desired entry or component with no way to reach it. Subjects in the *component-no history list* condition used the Go Back and go forward buttons to move along the precise path they had established; they could view each and every screen they had seen as presented in reverse order. Subjects in the *entry-no history list* were only presented with the default, text component of each entry they had accessed. From the text component, they still had to used the brass plate icons to find the correct component.

To understand how the GO BACK and GO FORWARD buttons were designed to work, imagine that a history list was kept for the subject though he or she never actually saw it. In the *component-no history list* condition, after each component was viewed, that component name was added to the top of the history list, much as is done with the *history list* tools. With the *entry-no history list* tool, each entry that was viewed was added to the top of the history list. As the subject pressed GO BACK, a pointer on the history list moved down the list and brought up the corresponding display. Under the *component-no history list* condition, the previous component screen was brought up and under the *entry-no history list* condition the previous entry's text (default) component was displayed. GO FORWARD correspondingly moved the pointer up the list. Figure 7 diagrammed the difference in backtracking jurisdictions as defined by the level factor- *entry* versus *component*.

The underlying history list did not cycle; that is, one could not GO BACK further than the bottom of the list and have the pointer loop back up to the top of the list. To help users understand this linear, non-cycling concept, the GO BACK and GO FORWARD buttons were greyed out and disabled when they did not apply. This technique was used to stress that when the bottom of the history list was reached, the GO BACK button had no effect. Similarly, when the subject pressed GO FORWARD and reached the top of the list, that button was disabled.

It is critical to remember that the GO BACK and GO FORWARD buttons did not actually modify or add to the history list; they merely replayed it. When the user ended the string of GO BACK and GO FORWARD commands, the "pointer" was moved to the top of the list and the list was then again appended. For the purposes of this experiment, this append-only history list was necessary to eliminate the problems involved in dealing with a branching, non-linear path.

One attribute common to all conditions was the way in which screen transitions were shown. When subjects were backtracking, the screen was displayed via a "wipe left." All other times, screens were displayed using a "wipe right." This was intended to help subjects develop an understanding of the mechanisms governing GO BACK (and GO FORWARD).

Also common to all four groups was equivalent backtracking functionalities. All groups had access to their entire history and could use GO BACK (and GO FORWARD) to move to any entry (or component) they had previously accessed.

Under test conditions, users were not required to navigate back more than seven entries. This determination was made in an effort to keep the number of tasks reasonable to maximize subject engagement and minimize fatigue. It was believed that the majority of backtracking would take place within this range.

For the *entry- history list* group, the portion of the history list that was displayed corresponded to the maximum length (seven entries) but offered a scroll bar to permit viewing of the entire history (see Figure 10). A conversion of 1-2/3 components per entry was used to determine the twelve-item display length for the *component- history list* group (see Figure 10). Assume that a piece of information has equal probabilities of being found in any of the three components. Merely by accessing an entry, a user automatically sees the text component. To view the table component, the user would invariably access the text first, then the table for a total of two components. Similarly, to access the figure, two components are viewed. Given the equal probability of requiring information from any of the three components, the expected number of viewed components is (1+2+2)/3 or 1-2/3 components per entry (assuming no errors). Using this information, the component level history list showed a window of twelve components while the entry level history list showed seven entries. As mentioned, both of these windows are scrollable, permitting access to the entire history list.

Backtracking Tasks. This section will summarize the two types of tasks that subjects completed: GO BACK and Choice tasks. The two are identical in sequence but differ in one respect. In the GO BACK tasks, subjects were required to use the backtracking tool while in the choice tasks, users could select the computer tool or manual (Table of Contents) backtracking. The sequence of events comprising a task is shown in Figure 11.

For this experiment, a task was defined as the entire sequence of interactions occurring between the time an anchor question (the initial question about the component that the GO BACK question refers to) appeared and the time the backtracking question was answered. A trial referred only to

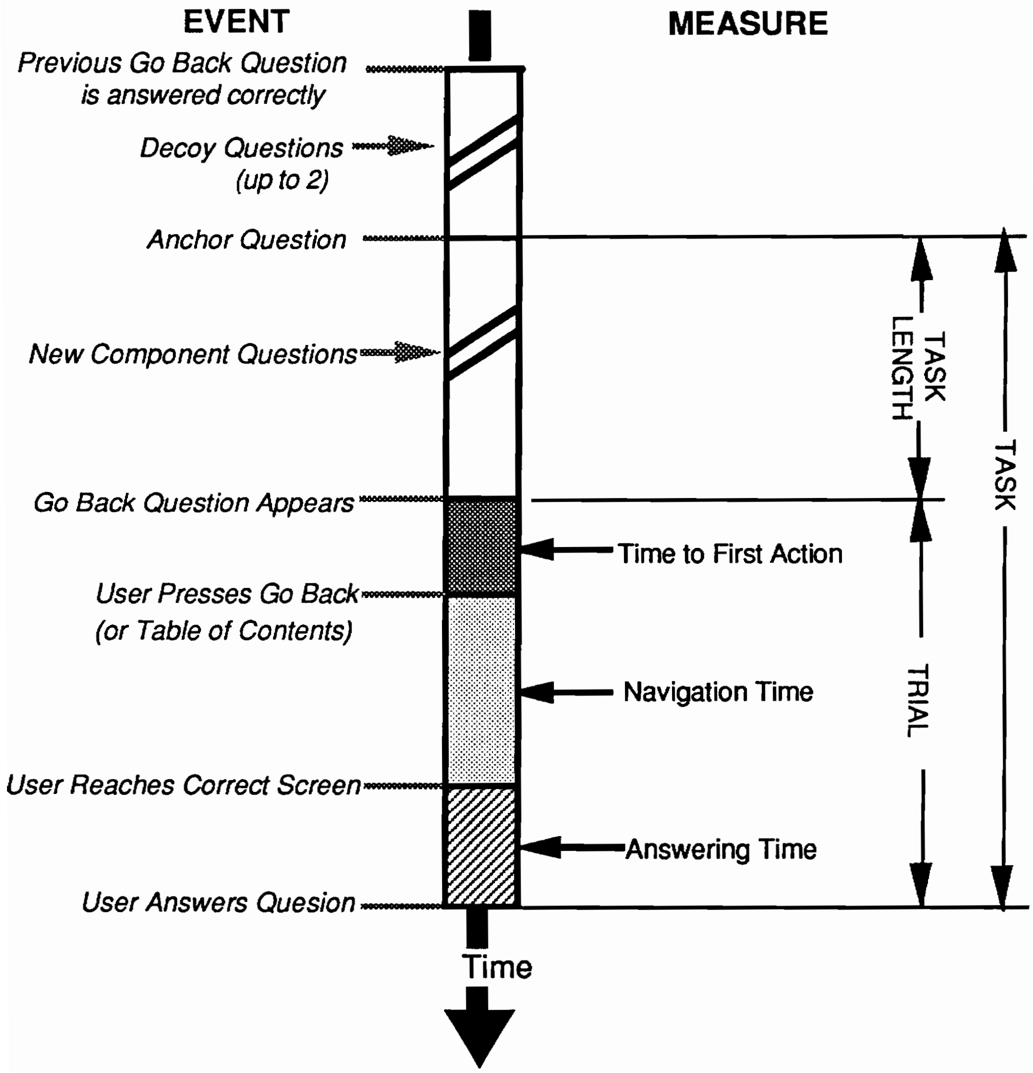


Figure 11. Relationship of events and measures in a task

the subject's actions from the time the GO BACK question appeared to the time it was answered.

Each task began with an *anchor question*; this is the question corresponding to a component that a GO BACK or choice trial would later refer back to. After the anchor question, a series of *new component* questions were asked to help form the history list for the subject. New component questions were always prefaced by "Use the TABLE OF CONTENTS to find .....".

Next, the actual trial question was asked. The GO BACK trial questions for the main study were prefaced by "Use GO BACK to find .....". The subjects were restricted to use the navigation method indicated. If they attempted to use the TABLE OF CONTENTS rather than GO BACK, they received the message "Please use the GO BACK button." Similarly, if they tried to press the GO BACK button for a new component question, they saw, "Please use the Table of Contents." Though the incorrect button was disabled, it was not greyed out.

The Choice trial questions were prefaced by "Find ...." and the subject had received instructions that he or she could use either the GO BACK button or the TABLE OF CONTENTS. In the group without computer-supported backtracking, both the GO BACK and Choice trials had the instruction "Use the TABLE OF CONTENTS to find ...."; this group did not have any other navigation option. The questions for all group were identical.

Questions were presented one at a time and tasks did not overlap. For instance, anchor question A and new component question B may be followed by GO BACK question C which is about component A. Next, anchor question D and new component questions E and F are followed by a second GO BACK question G. Questions must appear in the order A-B-C then D-E-F-G. Tasks do not overlap such as A-D-B-C-E-F-G. In this way, tasks and trials were kept separate so there was some degree of control over the contents of the user's history list.

Between tasks *decoy questions* could appear. These are merely filler questions to prevent the subject from assuming that the question immediately following a GO BACK or choice question is the anchor question. Up to two decoy questions were used between tasks; this number was kept small to maintain a tolerable session duration.

A total of five GO BACK and four Choice tasks were used in this study in addition to two GO BACK warm-ups and one Choice warm-up task. The tasks were contained within the 12 warm-up and 53 actual multiple choice test items. Each question had four choices provided with only one correct answer. The questions were very unambiguous requiring only information extraction and not interpretation. The entire list of questions and answers is given in Appendix E. Table 2 shows how the questions were grouped into tasks. The first question number indicates the anchor question and the last indicates the GO BACK or choice question. The W prefix on a question number indicates that it refers to a warm up task. The length is given in components.

Subjects were instructed that the computer would allow them to answer the question if and only if the correct component was in the display. If they attempted to answer without meeting that condition, perhaps from memory, they were given an error message which read "Please locate the appropriate card before attempting to answer this question." In addition, subjects had to answer the question correctly before continuing on. Incorrect answers resulted in the message "Incorrect. Please select another answer." Though subjects

TABLE 2

Sequence of Tasks and Numbering of Trials

Choice Trial Number	*Backtrack Trial Number*	*Questions*	*Length*	*Task Type*
		W1 - W4 *	4	GO BACK
WARM-UP		W5 - W7 *	3	GO BACK
		W8 - W12 *	5	Choice
	1.	1 - 3	2	GO BACK
1.		5 - 7	3	Choice
	2.	9 - 13	5	GO BACK
2.		14 - 18	6	Choice
3.		20 - 23	4	Choice
	3.	26 - 32	9	GO BACK
	4.	34 - 41	11	GO BACK
	5.	42 - 44	3	GO BACK
4.		46 - 53	10	Choice

* indicates warm-up question

had an unlimited number of opportunities to answer correctly, wrong answers were noted to check for poorly written questions or subjects who may have experienced difficulty. When the subject answered a question correctly, the screen flashed twice and the new question and bank of answers appeared.

Experimental Design

This study had two distinct, but interrelated, designs as shown in Figure 12. First, five groups (four treatment and one control) were used in a single factor between-subjects design to determine if the backtracking support had a performance advantage over no backtracking support. The five levels of the factor Group are:

- Group -- five levels -- between subject
  - *component-history list*
  - *entry-history list*
  - *component-no history list*
  - *entry-no history list*
  - *control*

Second, the four treatment groups (each with a different backtracking tool) were analyzed as a two factor between-subjects design with the following factors:

- Level (L) -- two levels -- between subject
  - backtracking tools navigate at *component* (lowest) level
  - backtracking tools navigate at *entry* (highest) level

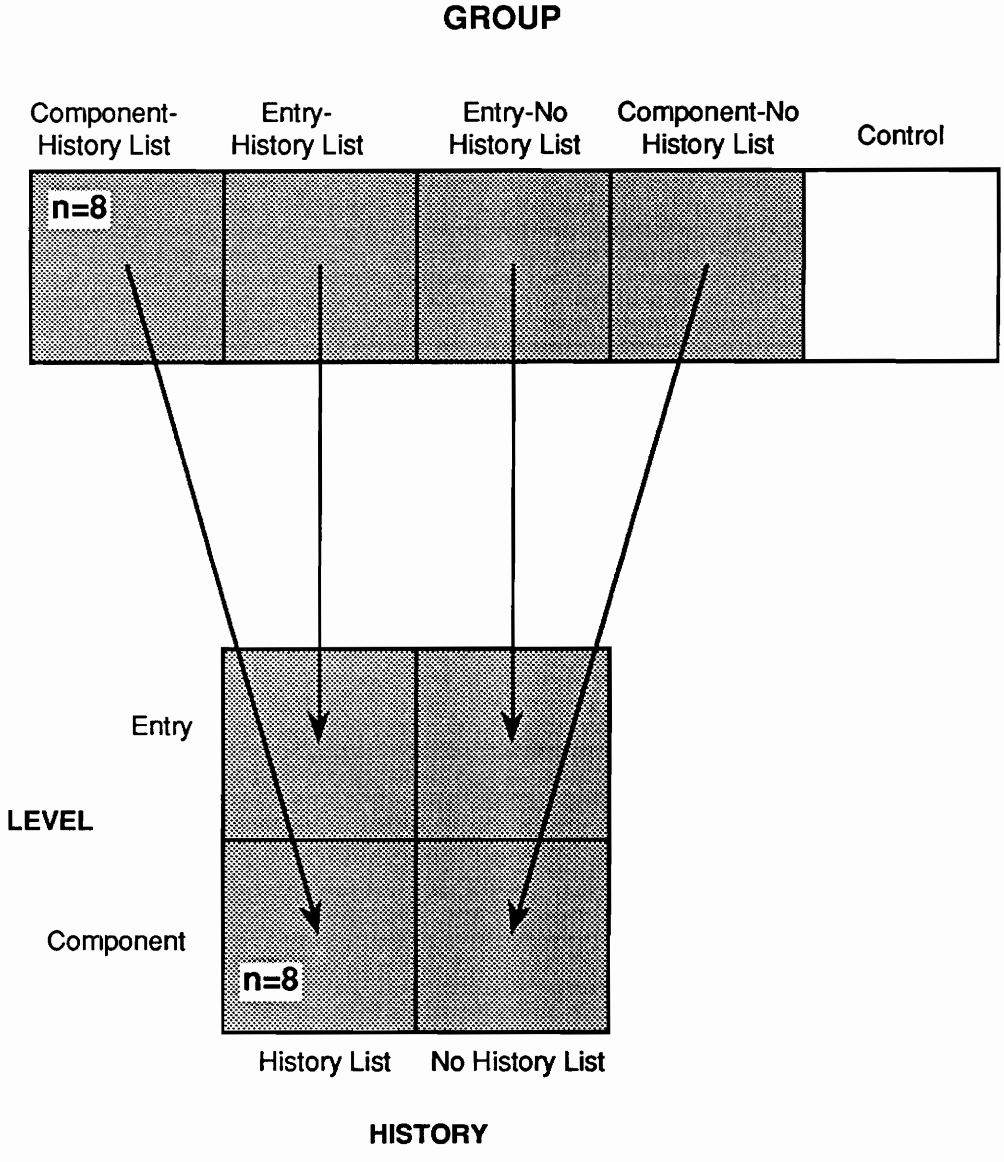


Figure 12. Relationship of one-factor and two-factor between subject designs.

• History (H) -- two levels -- between subject

- *history list* provided to directly select a destination node for direct navigation
- *no history list*, sequential backtracking through exact path taken through nodes

Four backtracking tools resulted from crossing the two factors. The four combinations were each used by 8 different subjects for a total of 32 subjects in the treatment conditions. The control group completed identical tasks without benefit of any backtracking support. This group always navigated by reaching an entry through the TABLE OF CONTENTS.

Subjects completed a total of three practice, five GO BACK, and three choice *Trials*. The number of components the subject was required to GO BACK in each trial was termed the *Task Length*. It was depicted in Figure 11. Task Length was randomized from two to eleven components. However, the sequence of questions and lengths was held constant across subjects.

The Task Lengths were determined by adding two to the first digit of each number in a sequence in a random number table. The last digit of the random number was used to select whether the task was a GO BACK task or Choice task. When the digit was odd it was the former and when even it was the latter.

Once the sequence and length of tasks was determined, the components from which the questions were extracted were drawn at random. If it had been determined that the task sequence included a Task Length of seven components, the components were drawn at random from a stack until they at least "equaled" the designated amount. Text components were assigned a value of one while figures and tables were assigned a value of two; this is because under the control condition (TABLE OF CONTENTS only) the text component had to be accessed before the figure or table could be accessed. The Task Length was permitted to increase by one when the last component drawn was a figure or table.

For one example, assume the last digit of a random number was odd and the first digit was nine. The last digit is odd so it was a GO BACK task. The first digit of the random number was nine so two were added (according to the algorithm) for a Task Length of eleven components. When drawing the components, ten points were gathered and the last component drawn was a figure or table. The resulting in Task Length was twelve components. As a result of the random length assignment and the random component drawing, the Task Length could therefore vary between two and twelve components.

Dependent Variables

Go Back Task. In the GO BACK task, the dependent measures were times from each of five GO BACK trials. Three times were extracted from each backtracking trial for each subject: Time To First Action, Navigation time, and Answering time. The relationship of these three times to specific task events was shown in Figure 11.

Time to first action indicates the time it takes a subject to press the GO BACK (or TABLE OF CONTENTS for the control group) button after being shown a backtracking question. This measure was used to determine if particular groups formed their action specification and executed it in less time than other groups. The second measure, Navigating time, began when subjects had pressed the GO BACK button (or TABLE OF CONTENTS for the control group) and concluded when the subject reached the correct screen needed to answer the question. The third measure, Answering time, was the time taken to scan the correct component screen and select the correct answer.

Choice Task. The Choice task was designed to record user preferences for each of the four backtracking tools as compared to the hierarchical TABLE OF CONTENTS. Interleaved with the GO BACK tasks were four opportunities for the user to re-access components by a method of his or her choosing: either the TABLE OF CONTENTS or GO BACK tool. The control group did not have the choice, but was still required to answer the questions to keep the tasks equal for both control and treatment groups. The dependent measure was the Frequency with which GO BACK is used.

The GO BACK and Choice tasks were not combined in an effort to keep the data as pure as possible. If users were always given a choice of using the GO BACK or the TABLE OF CONTENTS, there would probably be unequal sample sizes for the cells containing performance times for the GO BACK. In addition, the backtracking tasks were of unequal "length"; in some cases the subjects only had to refer back a few screens while in other cases they had to backtrack as many as twelve screens. This factor would undoubtedly confound the data. So although the data for the two studies was independent, the two types of tasks were interleaved in the experiment.

Questionnaire Responses. Questionnaire data was collected from each subject to gather demographic data and the subject's perceptions of the database navigation tools. The questionnaires for both the treatment and control groups are provided in Appendix D.

The first question pertained to how often the subject uses computers; this is the experience measure. For this question, it was assumed that the seven point scale is interval and represents the underlying continuum of exposure from no experience to daily experience. Questions two and three pertained to the subject's perception of overall navigation and the efficiency of the TABLE OF CONTENTS. Question four through six were given only to the treatment groups; they pertained to the functionality of GO BACK. Questions seven, eight, and nine were mainly exploratory in nature: what was perceived to be the best backtracking feature- entry versus component, history versus no history and the longest "length" that subjects would be comfortable backtracking.

Experimental Protocol

The subject was given two informed consent forms as shown in Appendix A. The first is a general form and approved for Logicon Technical Services, Inc. The second states the relationship of Virginia Tech to the study and provides further details specific to this study. The experimenter asked the subject to read each form and ask any questions he or she may have had. When the subject was finished, and all of his or her questions had been satisfactorily answered, he or she was given the choice to continue or to chose not to participate. If he or she agreed to continue, both subject and experimenter (witness) signed and dated the Logicon's consent form and the subject signed Virginia Tech's informed consent. Appendix B shows a statement about the liability insurance carried by Logicon that was made available if and when subjects inquired.

If the subject chose to participate and had signed the consent form, the subject was given a set of written instructions as shown in Appendix C. The first instruction sheet was given to subjects in treatment groups and the second sheet was given to subjects in the control group. After the subject read and signed the instructions, the experimenter provided a demonstration of the database showing how to access an entry via the TABLE OF CONTENTS. The experimenter provided an overview of the database content and explained, if necessary, about how the mouse worked. The experimenter showed how the component, the TABLE OF CONTENTS, and the GO BACK buttons functioned (for subjects in treatment groups). The subject was given time to explore the database and answer twelve sample questions using the TABLE OF CONTENTS and GO BACK (if applicable). The sample questions contained three backtracking opportunities - two GO BACK tasks and one choice task. After the subject answered all twelve questions, the database display read "You are now finished with the warm-up exercises." The experimenter asked again if there were any questions or difficulties. The subject was reminded that the questions are not trick questions and he or she should work quickly and accurately.

The subject began with the first question in the question display and the TABLE OF CONTENTS in the database display. The subject completed all 53 questions, then was given the questionnaire shown in Appendix D. The treatment groups received the first questionnaire with ten questions whereas the control group was given the second questionnaire with only four questions. The subject was told that the experimenter could help clarify questions if necessary. A subject code was recorded on the subject's questionnaire so that the data was properly classified by subject group. At the conclusion of the experimental session the subject was debriefed about the specific objective of the experiment.

RESULTS

GO BACK Trial Times

Effect of Group. The backtracking trial time data was analyzed to determine how each of the dependent measures (Time To First Action, Navigating, and Answering times) were affected by the factor Group. Two additional measures were included for analysis: Median Navigation time and Standard Deviation of Navigation time. Each subject's median Navigation time (third of five ranked Navigation times) was calculated to help minimize the effect that a few very large data points had when using the mean. The standard deviation statistic was calculated to determine if subjects in certain Groups had a significantly larger or smaller amount of time deviation. Each of the mean dependent variable times are shown in Appendix F, Tables 1 and 2. A comparison of the mean versus median Navigation time for each subject can be found in the scatter plot in Figure 13.

The effect of Group on each dependent variable was evaluated by a one-way ANOVA. Table 3 provides a summary of the F-ratios that were computed; the complete ANOVA summary tables can be found in Appendix G.

There were highly significant differences among groups for each of the dependent variables except Time to First Action and Standard Deviation. For each of the variables with a significant F-ratio, the Least Significant Difference (LSD) test was used for pairwise comparisons among the groups (See Table 4). Admittedly, it is the least stringent of the post-hoc tests and does not control for family-wide error. However, in this exploratory study, a type I error is

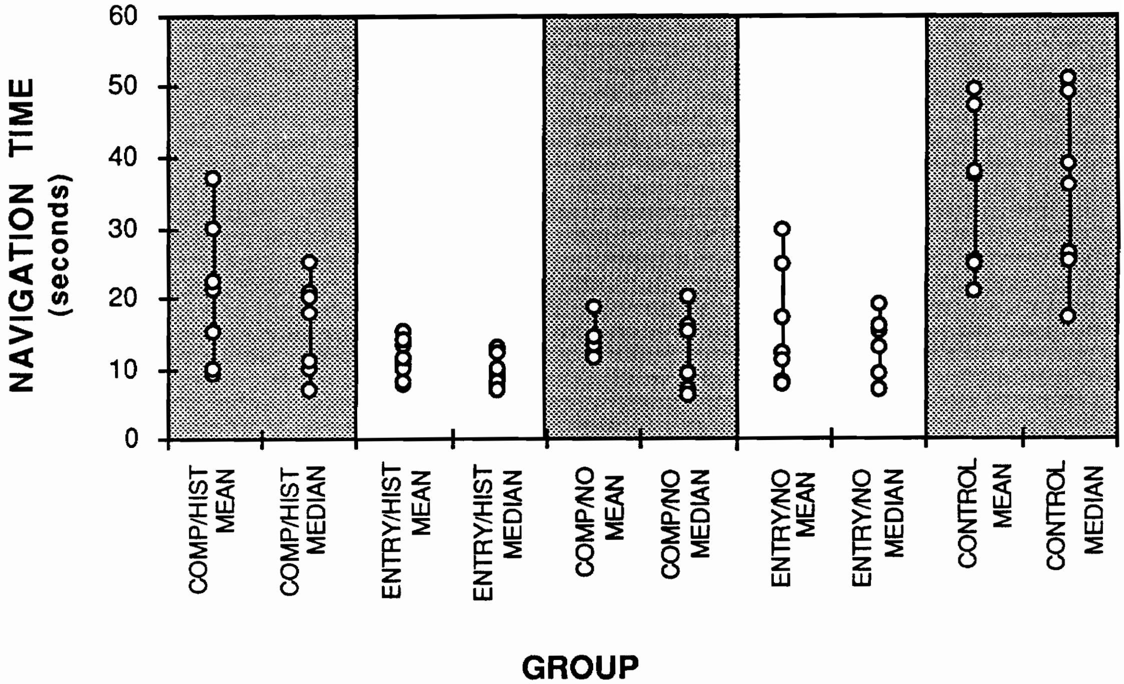


Figure 13. A comparison of the mean and median Navigation times for all subjects by Group.

TABLE 3

Computed F-values by Dependent Variable for the Factor Group

Dependent Variable	*F-ratio*
Time to First Action	2.37
Mean Navigating Time	9.69 ***
Median Navigation Time	15.79***
Answering Time	0.242
Standard Deviation of Navigation Times	1.52

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

TABLE 4

Differences between Groups as Determined by the LSD test for Mean Navigating Time. ("X" indicates a statistically significant difference at p<0.05)

GROUP	*Control*	*Component-History List*	*Entry-History List*	*Component-No History List*
Entry-No History List	X			
Component-No History List	X			
Entry-History List	X	X		
Component-History List	X			

preferable to a type II error. All possible differences are sought out and may be later verified in further studies.

Table 4 showed that subjects in the *control* group had significantly longer mean and median Navigating times than subjects in any of the treatment groups. Also, the *entry-history list* group had significantly shorter Navigation time than the *component-history list* group.

Effect of Trial. A post-hoc analysis was done to determine if subjects had significantly different Navigation times across Trials. That is, did some or all groups perform better on a particular trial (such as a trial with a small Task Length). To examine this, a two-way mixed-factor ANOVA, including both Group and Trial factors, was performed. The ANOVA summary table is included below in Table 5. Trial and Group x Trial both were significant effects in addition to the main effect of Group, which was established previously.

To further evaluate the Trial effect, the LSD test was used to determine significant differences between trials. The X's in Table 6 show which of these comparisons yielded significant differences. Significant differences existed between Trials 1 and 3, 1 and 4, 1 and 5, and 2 and 4.

Next, the interaction effect of Group x Trial was examined. Figure 14 shows the plot of the mean Navigation times for each Trial by Group. The LSD test was used for pairwise comparisons to help determine the nature of the interaction effect. A complete table of all significant differences found between combinations of Group and Trial may be found in Table 6 in Appendix G. To summarize the comparisons, out of the 300 total pairwise comparisons, 107 were significantly different, and of those 70 involved the control group.

TABLE 5

Two Factor ANOVA on Navigating Time

Source	***df***	***SS***	***MS***	***F***	***p***
Group	4	12098.200	3024.550	9.69	0.0001
Subj (Group)	35	10919.800	311.994		
Trial	4	3647.050	911.763	6.44	0.0001
Group x Trial	16	4402.250	275.141	1.94	0.0210
Trial x Subj (Group)	140	19814.700	141.534		

TABLE 6

Differences between Trials as Determined by the LSD test. ("X" indicates a statistically significant difference at $p<0.05$)

TRIAL	1	2	3	4
5	X			
4	X	X		
3	X			
2				

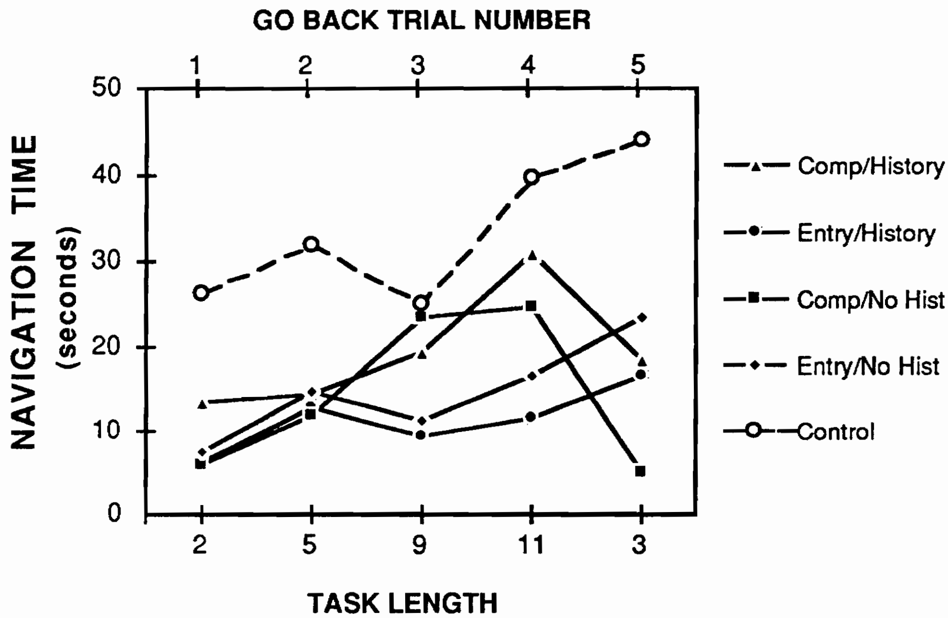


Figure 14. Mean Navigation time for each group on each backtracking trial

Comparisons between the treatment groups only produced 37 significant differences. Of those, 6 were within-group comparisons and 31 were between-group comparisons. Below is a description of some of the components contributing to the significant interaction effect of Group x Trial.

The first way to interpret Figure 14 is to examine differences in mean Navigation time across Trials within each Group. The LSD test showed that for each of the entry groups there were no significant differences in mean Navigation time between any of the trials. In other words, for both the *entry-history list* and *entry-no history list* groups, all trials resulted in essentially the same mean Navigation time regardless of the trial number or task length.

Differences did exist in the *component-history list* and *component-no history list* groups. In the *component-history list* group, the mean Navigation time for Trial 4 was significantly higher than for Trials 1 and 2. Recall that Trials 1 and 2 had relatively short Task Lengths of 2 and 5 components respectively while Trial 4 had the longest Task Length of 11 components. The same type of pattern appears in the *component-no history list* group. Trial 1, the shortest trial (Task Length = 2 components) and Trial 5 (Task Length = 3) had significantly smaller mean Navigation times than either of the two longest trials: 3 (Task Length = 9) and 4 (Task Length = 11).

For the control group, times on Trials 1 and 3 were each significantly faster than Trials 4 and 5. This indicates a generally increasing trend in mean Navigation time over Trials.

In sum, the *entry* groups show no change over trials. The control group demonstrates a general increase from Trials 1 and 2 to 4 and 5. The *component-history list* group increases to a peak at Trial 4 while the component-no history list group peaks at both Trials 3 and 4 with a significant drop to Trial 5.

The second way to interpret Figure 14 is to examine differences in mean Navigation time across Groups within each Trial. The plot shows a general divergence of the means as the trials progressed. For Trials 1 and 2, there were no between treatment group differences, only differences between the control group and three of the treatment groups. For Trial 3, there was one between-treatment group difference (*component-no history list* vs. *entry-history list*); for Trial 4, three between-treatment group differences (*entry-no history list* vs. *component-history list, entry-history list* vs. *component-history list,* and *entry-history list* vs. *component-no history list*); and for Trial 5, two between-treatment group differences (*entry-no history list* vs. *component-no history list* and *component-history list* vs. *component-no history list*).

The characteristics of the group means from trial to trial are difficult to describe succinctly and without exception. In general, with the relatively short tasks (Trials 1 and 2) there are no significant performance differences between groups. When considering the trials with longer task lengths (Trials 3 and 4), the *entry-history list* group has a significant navigation advantage over the *component-no history list* group. Lastly, trial 5 has a short task length (3 components) but does not demonstrate the tight grouping of means as seen with Trials 1 and 2.

Effect of Level and History. This section discusses the specific effects of Level, History, and Level x History on the dependent variable Navigation time. Table 7 shows the F-ratios associated with each dependent variable that was found significant in the previous section, mean and median

TABLE 7

Computed F-values for Effect of Level, History, and Level x History on Navigating Time

	FACTOR		
Dependent Variable	*Level*	*History*	*Level x History*
Mean Navigating Time	2.38	0.11	2.99
Median Navigating Time	1.13	0.07	4.67*

* $p < 0.05$

Navigating time. Neither the Level nor History Factors had any significant effect on any of the dependent variables. The median Navigation time showed an interaction effect which is illustrated in Figure 15.

As discussed in a previous section, the only significant difference among treatment groups as determined by the LSD test was the comparison of the *entry-history list* and *component-history list* groups. However, neither of these two groups is statistically different from the *entry-no history list* or *component-no history list* groups.

Choice Frequencies

Neither Level nor History nor the interaction had a significant effect on the choice frequency data collected in this study. The data is available in Table 3 in Appendix F and the ANOVA summary table is available in Table 9 in Appendix G.

Questionnaire Responses

The questionnaires that will be discussed in this section are provided in Appendix D. The factor Group had no effect on mean responses to questionnaire items one through eight.

Question one dealt with computer experience. An ANOVA on the factor Group showed that the F-ratio was insignificant ($p<0.80$) and groups did not differ in reported computer experience. Question numbers two through seven offered ordinal response selection and were evaluated with the Kruskal-Wallis

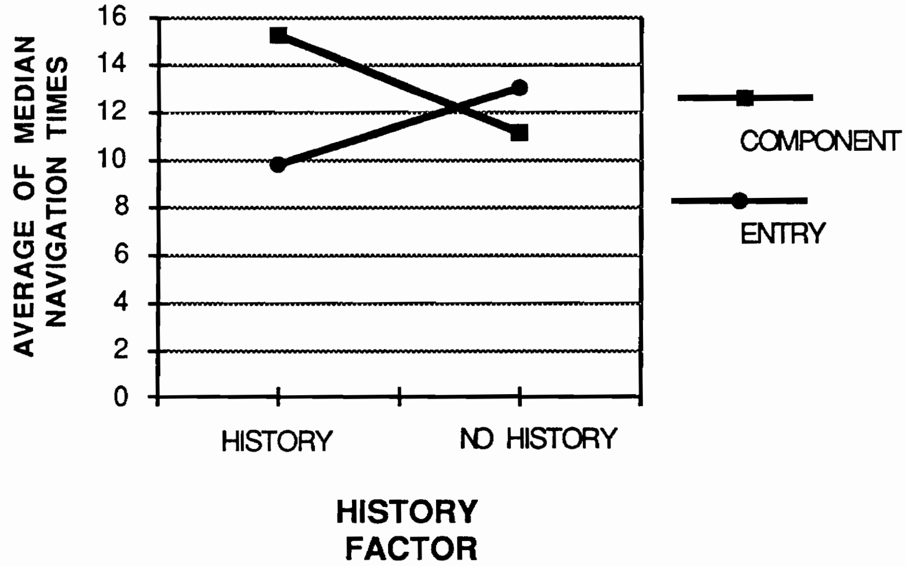


Figure 15. Interaction effect of Level and History on Median Navigation Time

one-way analysis of variance test. The analysis revealed no significant effect of Group ($p>0.05$).

Paired t-tests were used to determine if subjects rated the efficiency of the TABLE OF CONTENTS (item 3) significantly differently than the efficiency of GO BACK (item 4). Table 8 shows the results of the test. It indicates that there was a significant difference only within the *entry-history list* group. They rated the computer GO BACK tool as significantly more efficient than the TABLE OF CONTENTS.

Question numbers eight and nine each had just two possible responses and were administered to the treatment groups only. Because the responses yielded categorical data, the Fisher Exact test for 2x2 tables was used for pairwise comparisons between groups. It revealed a difference in mean responses to item nine. This question asked subjects if they would prefer a history list or no history list. The *entry- history list* group (all preferring history list) responded significantly differently ($p=0.04$) from the *entry-no history list* group (only four preferring history list). Neither of these groups responded significantly differently from the *component-history list* group (seven preferring history list) or the *component-no history list* group (five preferring history list).

Correlations

Correlations between the following variables were calculated:

- Task Length and Mean Navigation Time
- Mean Navigation Time and reported computer experience
- Mean Navigation Time and questionnaire responses two through five

TABLE 8

Student's t-tests Between Rated Efficiency of the Table of Contents (questionnaire item 3) and the Rated Efficiency of Go Back (item 4)

Group	*Mean Difference*	*t*	*p*
Component/History List	-0.625	-0.576	0.5825
Entry/History List	-2.625	-4.406	0.0031
Component/No History List	-1.500	-1.821	0.1114
Entry/No History List	-0.875	-1.263	0.2470

- Choice Frequency and reported computer experience
- Choice Frequency and questionnaire responses two through five

Originally, both the Pearson correlation coefficient (R) and the Spearman rank-order correlation coefficient (r_s) were used to calculate measures of association. Whenever there was doubt as to whether one or both sets of data were interval scale, the Spearman rank-order correlation coefficient (non-parametric) was calculated. As mentioned, there is no definitive answer as to whether or not many of the questionnaire scales were interval. However, both calculations resulted in nearly the same coefficients and p-values. In all cases, the Pearson correlation coefficient was more conservative than the Spearman rank-order coefficient. For brevity, only Pearson's R will be reported.

Correlations were calculated for both the mean and median Navigation times. There was not a great deal of difference between the two but in general the mean Navigation time was a stronger correlate. Therefore, only the correlations with mean Navigation time are reported.

Previous analysis indicated that Trial and Trial x Group had effects on mean Navigation Time. To examine that further, possible correlations were investigated between task length and mean navigation time. This correlation was not significant across all subjects. Within groups, only the *component-no history list* group had a significant correlation with R=0.787 and p= 0.0001.

Table 9 shows all the correlations between mean Navigation time and questionnaire responses and between choice frequency and questionnaire responses. These correlations are taken across all subjects which in some cases meant only across all treatment subjects. This was because control

TABLE 9

Correlation Coefficients between Questionnaire Responses and Mean Navigation Time and Choice Frequency across all subjects.

Questionnaire Item	***Navigation Time***	***Choice Frequency***
1. Computer Experience	-0.3137[1] *	0.2509[2]
2. Ease of Navigation	0.0876[1]	-0.0013[2]
3. Efficiency of Table of Contents	0.3431[1] *	-0.0336[2]
4. Efficiency of Go Back	0.0146[2]	0.3785[2] *
5. Ease of Backtracking	-0.3515[2] *	0.3279[2]

* $p < 0.05$

[1] - indicates measure is across all 40 subjects [2] - indicates measure is across 32 treatment subjects

subjects did not have the choice frequency measure or questionnaire items four through nine.

Mean Navigation time was significantly correlated with questionnaire items one, three, and five. That is, for a low mean Navigation time (i.e. good performance), reported computer experience (item one) tended to be high, the TABLE OF CONTENTS tended to be rated as inefficient (item three), and GO BACK was rated as easy to use (item five). The converse holds true: high Navigation time, little computer experience, found TABLE OF CONTENTS to be efficient and found GO BACK to be difficult to use.

The choice Frequency correlated significantly with questionnaire item four: the efficiency of GO BACK. That means subjects who used Go Back more often tended to rate it as more efficient. This offers some support for the notion that all subjects may not have had enough experience with the backtracking tool to really determine its effectiveness.

There is a significant correlation (R=-0.447) between choice frequency and mean Navigation time across all 32 subjects. Only the entry-*no history list* group had a significant (p=0.007) within-group correlation (R=-0.851).

DISCUSSION

Benefits of Backtracking

The mean and median Navigating times were all significantly affected by the Group factor. Navigating time has considerable face validity as a measure of backtracking performance for each group. It does not include the initial action specification time or the time required to view a component and select the correct answer. It only represents the length of time needed to locate a component once the sequence of actions has been initiated. Therefore, Navigation time appears to be a suitable indicator of backtracking performance time.

The control group was significantly slower than any of the treatment groups. To backtrack, the *entry-history list* group required, on average, only 34% of the time needed by the control group. Even the slowest treatment group, the *component-history list* group, took only 57% of the control group's time. Clearly, any computer-supported backtracking tool results in a significantly smaller navigation time than no backtracking tool.

The control group demonstrated an interesting phenomenon of a general increase in Navigation time over time for each of the backtracking trials. They did not show an expected learning effect. Subjects in the treatment groups may have suffered less from fatigue and frustration than subjects in the control group because of the intermittent reprieve from the tedious TABLE OF CONTENTS navigating. The increase in Navigation time cannot be ignored by interface designers because user frustration is a real concern.

The reduction in Navigation time with computer-supported backtracking tools confirmed several general statements made by researchers about user recovery. Archer, Conway, and Schneider (1984) noted that manual recovery is slow and automated recovery is more efficient and effective. Leeman (1986) stated how important computerized undos are for human factors design. The survey by Yang (1989) revealed that people feel that recovery is useful and beneficial. Indeed these statements are not definitive about user recovery. However, in the case of database navigation, this research supported each of these statements.

Comparison of Backtracking Tools

Navigation time is intended to be used as an estimate of backtracking performance. However, a few subjects had one inordinately large number among their five times. These were due to a mistake such as by-passing a component or entry (in *no history list* groups) or a disorientation caused by accidentally selecting an incorrect entry (in *history list* groups). Because each subject contributed only five data points to this dependent variable, one large time could substantially affect the mean. To offset this, the median Navigation time was analyzed as a possible measure of the underlying backtracking performance construct.

The effect of Level x History was significant on median Navigation time. However, Level x History did not significantly affect the mean Navigation time. Because one measure cannot be "proven" to be more or less valid than another, differences in median Navigation time cannot be considered an incontestable effect of Level x History.

Only the *entry-history list* and *component-history list* groups were significantly different even with a very liberal test (LSD). On average, subjects in the *entry-history list* group took only approximately 59% as long to backtrack as subjects in the *component-history list* group.

There were no differences within or between the *no-history list* groups. One explanation for this can be found with the mental model concept as described by Hutchins, Hollan, and Norman (1986). They recommend reducing the cognitive workload by closely matching the user's mental concept of a system with the actual appearance and operation of a system. Without a *history list*, users receive direct feedback through their actions. They are able to see the screen wipe horizontally as if they are flipping backward through pages. Regardless of the level (*component* or *entry*), users' performance times are equivalent overall without a history list.

When supplied with a *history list*, a mental model must be created and a user must reestablish a sense of position within the database. Perhaps this model is developed most quickly at the *entry* level. A user need only scan a list for the proper state name and select the component from the "brass plate". The subject is used to selecting the component this way because that is how it is done when accessing an entry from the TABLE OF CONTENTS as well. Conversely, with the *component-history list* group, users had to learn another way of selecting components i.e. from the history list. Also, the history list is longer and more complex than for the entry-history list group and the component is selected under some uncertainty. This process may have taken longer than the simple search and select strategy used with the short *entry-history list*. Perhaps when providing a history list, it is wise to avoid making it overly detailed to bog down a user during backtracking.

A range of task lengths from two to eleven components was used in this experiment. This was done to prevent subjects from counting screens or using other artificial techniques to aid in backtracking. Both Trial and Trial x Group had significant effects on mean Navigation time. This does not imply that actual task length had an effect but merely the specific trial had an effect. The correlation between task length and Navigation time was insignificant across all groups. However, this correlation was significant within the *component-no history list* group. This group had significantly longer Navigation times for the trials with long task lengths (Trials 3 and 4) than the trials with short task lengths (Trials 1 and 5). This is not a very surprising result given the tool's functionality. Subjects had to navigate back through each component sequentially. The longer the Task Length, the more screens the user had to pass through. The *entry* groups did not have a significant correlation between Navigation time and Task Length and showed no differences across Trials. If an interface designer's goal is to facilitate a consistent backtracking time regardless of task length, the subjects using the *entry* tools demonstrated this homogeneity.

The questionnaire data showed no differences between groups for ratings of the navigation tools (items one through seven). Despite some differences in mean and median Navigation time, the groups reported the same average ratings. That is, even though there were performance differences, subjects did not perceive the tools as more or less efficient than the others. However, in this design, subjects did not have the opportunity to interact with the other backtracking tools. Their reported ratings are on an absolute rather than relative scale. However, a comparison of the ratings on the efficiency of the TABLE OF CONTENTS (item three) and the efficiency of the computer tool (item four) showed that the *entry-history list* group rated the tool as more efficient than manual backtracking.

The comparison of backtracking tools produced a contradiction to Field and Apperley's (1990) finding that an explicit history list results in a lower search time than no explicit history list. No differences were found in this study as a result of the History factor. Their study was done in the context of a menu search task and may not be generalizable to a database search task.

Future Research

Several extensions to this research are necessary to advance knowledge in this area. The first extension speaks to the main weakness of this research: the number of subjects as well as the number of trials was limited. The number of trials was significantly impacted by the desire to keep tasks separate and not interwoven. That was done to ensure equivalent history lists for each subject rather than risk having them accumulate unique history paths. A control should be developed for the history lists so the tasks may indeed be interleaved and more data could be extracted.

Second, this study should be extended to include a wider range of Task Lengths. The range used for this study was quite limited: 2 to 11 components.

User strategies may change as the Task Length approaches 20 or 25 components.

Third, the backtracking tools for this database operated only on entry and/or component nodes. This was a contradiction to both Smith and Mosier's (1986) recommendation that *any* user action should be undoable and Thimbleby's (1990) suggestion that the recovery tool be *unrestricted* in domain. If these guidelines were followed, the backtracking tools could have theoretically taken the user to the TABLE OF CONTENTS node! In this study, subjects with the backtracking tools showed a significant decrease in Navigation time despite this guideline violation. This research can neither reject nor support these guidelines. A follow-up study should be conducted to include additional groups who are provided with unrestricted backtracking.

Fourth, the effect of Task Length and Task Length x Group on Navigating time is not completely understood. There may be a need, in large complex databases, to provide a backtracking tool for "short" backtracks as well as one for "long" backtracks. The range of task lengths included in the experiment must be expanded to include lengths of up to perhaps 20 or 25. This research may also reveal a point at which backtracking becomes less efficient, in terms of time, than other navigation methods. With this information, users could be instructed when to use a backtracking tool for maximum efficiency. This would eliminate the need for the user to form his or her own model of when to use particular tools.

Fifth, this research addressed only some very high-level questions about backtracking and many details have not been investigated. The features of the backtracking tools were selected, in general, for simplicity. This linear undo/redo framework used for the tools should be compared to other methods such as tree structures and graphical representations. The role of the user's memory capacity should be investigated as it related to when a user chooses computer-supported backtracking. The item arrangement (chronological, alphabetical) needs to be empirically determined for a variety of situations.

This area of research is new and well deserving of attention by the human factors community. Users want computerized databases to offer flexible but simple navigation tools that are a help and not a hindrance.... and details seemingly as small as a backtracking interface can often determine the success or failure of software.

Conclusions

This study has effectively shown the value of backtracking support in a hierarchical database. This backtracking support, regardless of the level (*component* or *entry*) or if there is a history list, provides a significant time savings over no backtracking. Without exposure to all tools, users do not perceive a single tool as significantly easier (or more difficult) or more efficient (or less efficient) to use.

Though a specific recommendation cannot be made with respect to a single backtracking tools, several general guidelines can be offered. If an interface designer's goal is to facilitate a consistent navigation time, he or she should consider implementing a system which operates at an *entry* rather than a *component* level. This will enable the user to have equivalent backtracking times despite the Task Length.

If a designer chooses to incorporate a history list, he or she should avoid creating a lengthy, overly detailed *component-history list*. Navigation time is reduced for an *entry* versus a *component-history list*.

The above recommendations can be applied to CASHE:HPVS. The experimental database was designed very similarly to CASHE:HPVS and the backtracking task was not unlike one that a CASHE:HPVS user would encounter. Ultimately, designers of this large database should provide some type of backtracking support rather than requiring users to backtrack manually with an index. The above guidelines may be cautiously applied to the backtracking tool with the knowledge that there is no evidence that users perceive a performance difference.

REFERENCES

Archer, J. E., Conway, R., and Schneider, F. B. (1984). User recovery and reversal in interactive systems. *ACM Transactions on Programming Languages and Systems, 6* (1), 1-19.

Beard, D. V., and Walker, J. Q. (1990). Navigation techniques to improve the display of large two-dimensional spaces. *Behaviour and Information Technology, 9* (6), 451-466.

Boff, K. R., and Lincoln, J. (Eds.) (1988) *Engineering data compendium: Human perception and performance.* Wright-Patterson AFB, OH: Armstrong Aerospace Medical Research Laboratory, AAMRL/NATO.

Boff, K. R., Monk, D. L., Swierenga, S. J., Brown, C. E., and Cody, W. J. (1991). *Computer-aided human factors for system designers.* Human Factors Society 35th Annual Meeting. San Francisco, CA: Human Factors Society.

Canter, D. (1977). *The psychology of place.* New York: St. Martin's Press.

Canter, D., Powell, J., Wishart, J., and Roderick, C. (1986). User navigation in complex database systems. *Behaviour and Information Technology, 5* (3), 249-257.

Canter, D., Rivers, R., and Storrs, G. (1985). Characterizing user navigation through complex data structures. *Behaviour and Information Technology, 4* (2), 93-102.

Department of Defense (1989). *Human engineering design criteria for military systems, equipment and facilities (MIL-STD-1472D).* Washington, D.C.

Field, G. E., and Apperley, M. D. (1990). Context and selective retreat in hierarchical menu structures. *Behaviour and Information Technology, 9* (2), 133-146.

Fitter, M. (1979). Towards more 'natural' interactive systems. *International Journal of Man-Machine Studies, 11*, 339-350.

Greenberg, S., and Witten, I. H. (1988). How users repeat their actions on computers: principles for design of history mechanisms. In *Proceedings of CHI '88* (pp. 171-178). Washington, D.C.: ACM.

Hutchins, E. L., Hollan, J. D., and Norman, D. A. (1986). Direct manipulation interfaces. In D. A. Norman, and S. W. Draper (Eds.), *User centered system design* (pp. 87-124). Hillsdale, NJ: Lawrence Erlbaum Associates.

Leeman, G. B. (1986). A formal approach to undo operations in programming languages. *ACM Transactions on Programming Languages and Systems, 8* (1), 50-87.

Lincoln, J. E., and Boff, K. R. (1988). Making behavioral data useful for system design applications: Development of the Engineering Data Compendium. In *Proceedings of the Human Factors Society 32nd Annual Meeting* (pp. 1021-1025). Santa Monica, CA: Human Factors Society.

McMath, C. F., Tamaru, R. S., and Rada, R. (1989). A graphical thesaurus based information retrieval system. *International Journal of Man-Machine Studies, 31* (2), 121-147.

Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81-97.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman, and S. W. Draper (Eds.), *User centered system design* (pp. 31-61). Hillsdale, NJ: Lawrence Erlbaum Associates.

Quillian, M. R. (1968). Semantic memory. In M. Minsky (Ed.) *Semantic information processing*. Cambridge, MA: MIT Press.

Shneiderman, B. (1987). *Designing the user interface*. Reading, MA: Addison-Wesley Publishing Company.

Siegel, S., and Castellan, N. J. (1988). *Nonparametric statistics for the behavioral sciences*. New York: McGraw-Hill.

Smith, S. L., and Mosier, J. N. (1986). *Guidelines for designing user interface software* (Tech. Report ESD-TR-86-278). Bedford, MA: MITRE Corporation.

Thimbleby, H. (1990). *User interface design*. New York: ACM Press/Addison Wesley.

Toriya, H., Satoh, T., Ueda, K., and Chiyokura, H. (1986). Undo and redo operations for solid modeling. *IEEE Computer Graphics and Applications, 6*, 35-42.

Vitter, J. S. (1984). US&R: a new framework for redoing. *IEEE Software, 1* (4), 39-52.

Weiser, M. and Shneiderman, B. (1987). Human factors of computer programming. In G. Salvendy (Ed.), *Handbook of human factors* (pp. 1398 - 1415). New York: John Wiley and Sons, Inc.

Williges, R. C., Williges, B. H., and Elkerton, J. (1987). Software interface design. In G. Salvendy (Ed.), *Handbook of human factors* (pp. 1415 - 1449). New York: John Wiley and Sons, Inc.

Yang, Y. (1988a). A new conceptual model for interactive user recovery and command reuse facilities. In *Proceedings of CHI '88* (pp. 165-170). Washington, D.C.: ACM.

Yang, Y. (1988b). Undo support models. *International Journal of Man-Machine Studies, 28*, 457-481.

Yang, Y. (1989). Survey steered design: Evaluating user recovery and command reuse support by questionnaire. *Behaviour and Information Technology, 8* (6), 437-459.

APPENDIX A

Participant's Informed Consent

CONSENT FORM

TITLE: Design Effectiveness Research (71841221)

1. This effort is designed to examine the processes involved in system design, and to develop and evaluate design support systems. The purpose of this design effectiveness research is to increase the consideration of human performance data by system designers, thereby leading to increases in system usability. Studies will typically require you to use paper- and computer- based design aids to develop and evaluate designs. You will be asked to participate in five to ten experimental sessions, with each session lasting one to two hours. An overall participation time of approximately 10-15 hours, per subject, is anticipated.

2. Each experiment will involve the use of one or more design aids in the design and/or evaluation of a system. Design information will be presented either through (a) volumes of human performance data printed on 8 1/2 by 11 inch sheets of paper, or (b) microcomputer-controlled CRT displays. The purpose of these procedures is generally to assess your performance with and without the use of design aids, and to observe your use of design information.

3. The foreseeable risks associated with these experiments are comparable to those of viewing a standard television monitor for one hour at a time. You should experience minimal to no discomfort.

4. An expected benefit of this research is that our understanding of how to impact and increase the consideration of human performance data by system designers, with the ultimate goal of improving system usability, will be significantly enhanced.

5. Questionnaires and structured interviews are alternative procedures which will be used to compliment these laboratory studies.

6. I, ____________________________________, am participating because I want to. The decision to participate in this research study is completely voluntary on my part. No one has coerced or intimidated me into participating in this program.

Kenneth R. Boff or Sarah J. Swierenga adequately answered any and all questions I have asked about this study, my participation, and the procedures involved, which are set forth above, which I have read. I understand that the Principal Investigator or his designee will be available to answer any questions concerning procedures throughout this study. I understand that if significan new findings develop during the course of this research which may relate to my decision to continue participation, I will be informed. I further understand that I may withdraw this consent at any time and discontinue further participation in this study without prejudice to my entitlements. I also understand that the Medical Consultant for this study may terminate my participation in this study if he/she feels this to be in my best interest. I may be required to undergo certain further examinations, if in the opinion of the Medical Consultant, such examinations are necessary for my health or well being.

7. I understand that my entitlement to medical care or compensation in the event of injury are governed by federal laws and regulations, and that if I desire further information I may contact the Principal Investigator.

____

Volunteer Signature

8. I understand that for my participation in this project I shall be entitled to payment as specified in the DOD Pay and Entitlements Manual or in current contracts.

9. I understand that my participation in this study may be photographed, filmed or audio/videotaped. I consent to the use of these media for training purposes and understand that any release of records of my participation in this study may only be disclosed according to federal law, including the Federal Privacy Act, 5 5 U.S.C. 552a, and its implementing regulations. This means personal information will not be released to an unauthorized source without my permission.

10. I FULLY UNDERSTAND THAT I AM MAKING A DECISION WHETHER OR NOT TO PARTICIPATE. MY SIGNATURE INDICATED THAT I HAVE DECIDED TO PARTICIPATE HAVING READ THE INFORMATION PROVIDED ABOVE.

Volunteer Signature and SSN   Date

Witness Signature   Date

Principal Investigator Signature   Date

INFORMATION PROTECTED BY THE PRIVACY ACT OF 1974

Authority: 10 U.S.C. 8012, Secretary of the Air Force; powers and duties; delegation by: implemented by DOI 12-1, Office Locator.

Purpose: is to request consent for participation in approved medical research studies. Disclosure is voluntary.

Routine Use: Information may be disclosed for any of the blanket routine uses published by the Air Force and reprinted in AFP 12-36 and in Federal Register 52 FR 16431.

PARTICIPANT'S INFORMED CONSENT

The purpose of this study is to examine methods for navigating within a computerized database. After learning how to use the database, you will be asked to answer a series of 65 multiple choice questions. At the conclusion of the exercise, you will be given a very brief questionnaire to complete pertaining to your opinion of the navigation tools in the database.

It is expected that your participation will take approximately one and one-half hour. It should take about fifteen minutes to become familiar with the database, one hour to complete the exercise, and ten minutes to fill out the questionnaire. You will be provided with a break if needed at any time.

This results of this study will be used for a Master's thesis in Industrial and Systems Engineering at Virginia Polytechnic Institute and State University. The ISE research team members include:

Cortney G. Vargo, Masters Student (513) 476-7142
Dr. Robert C. Williges, Faculty Member (703) 231-6270

If you decide to participate, you will be paid $5 per hour or each portion of an hour. Therefore, you will receive $10 for one and one-half hour of your time.

There are no known risks associated with this research. There is only the possibility of fatigue from the length of the experimental session or from viewing the computer monitor.

As a research participant, you have certain rights:

1. It is your right to chose not to participate or to discontinue participation at any time for any reason.
2. Members of the research team will answer all questions concerning the research. In such cases when a complete answer could jeopardize the validity of the data, thorough answers will be delayed, with your permission, until the conclusion of the session.
3. Data collected from your participation will be treated with anonymity and confidentiality.
4. You have the right to be aware of any benefits or risks that may come about as a result of your participation.
5. If any further questions arise, you may contact either of the research team members listed above. You may also contact Dr. E. R. Stout, Chairman of the Institutional Review Board at Virginia Tech. He oversees the rights of participants in research studies and his number is (703) 231-5281.

Before you sign this form, be sure you understand, to your complete satisfaction, the nature of this study and your rights as a participant. Please feel free to ask any questions you may have at this time. If you decide to participate, sign below. Your participation is greatly appreciated.

CONSENT: I have read this consent form and understand my rights. I hereby agree to participate, but understand that I can withdraw at any time.

____

Volunteer Signature and SSN Date

APPENDIX B

Liability Insurance Coverage

LOGICON INTEROFFICE CORRESPONDENCE

TO: Air Force Human Use Review Committee
FROM: George Bermudez
SUBJECT: Experimental Subjects

DATE: September 25, 1989
FR-670-90
DISTRIBUTION

1. Regarding Logicon Technical Services, Inc. (LTSI) liability for injury to human subjects used in conducting Air Force experiments:

   a. It is LTSI's responsibility to ensure that the health and safety of the subjects is protected and that the subject is aware of and understands the risks involved...and has accepted those risks.

   b. The subject will be covered should he/she be injured in performing assigned duties, under Ohio's Worker's Compensation program.

   c. Logicon also carries employer's liability, comprehensive general liability and automobile liability insurance. Certificates of insurance will be provided upon request.

2. Human subjects provided by LTSI are typically recruited from local universities, work part-time in no risk or low risk experiments, and start at an hourly rate of $5.00. After several years of training and experience, their salary may increase to approximately $6.00 per hour. This ensures that there is not a financial inducement for participation in experiments that may involve some risk.

APPENDIX C

Experimental Instructions

EXPERIMENTAL INSTRUCTIONS

The purpose of this experiment is to collect information regarding navigation in a computerized database. You are asked to complete one session taking no more than two hours.

First, you will be given an introduction to the database. This database contains information about the states in three imaginary countries: Macaye, Kinsela, and Strathmore. Each country is composed of four provinces which are in turn composed of four states. Every state is represented by an "entry." Each entry is made up of three components: *text, table,* and *figure.* The *text* component gives information on the people and culture of the state. The *table* contains information about the state's economics and commerce. Lastly, the *figure* provides data on weather and the land. You are interested in vacationing in one of the states but you have not decided which one. You have come to the travel agent with a list of 65 questions. You are given access to the database to answer the questions sequentially.

Your experimenter will explain the two methods of finding entries of information - the TABLE OF CONTENTS and the GO BACK buttons. You will be shown how to access text, table, and figure components. You will be given 12 sample questions to answer to give you practice with the database. Feel free to ask your experimenter any questions that you wish at this time.

You will then be asked to complete the remaining 53 multiple choice questions pertaining to information that can be found within the database. You may ask questions at any time but the experimenter reserves the right to delay some detailed explanations until the conclusion of both experimental sessions.

Each question will begin with a phrase that tells how to navigate to the component to answer each question. Some will say *"Use the TABLE OF CONTENTS to find..."*, others will say *"Use GO BACK to find ...."*, while the remainder will begin only with *"Find...."*. In the third case, the method is not explicitly specified; you may use the TABLE OF CONTENTS or GO BACK (if applicable). Try to answer the questions as quickly and accurately as possible. They are not trick questions! After you have completed the 65 questions, you will be asked to complete a brief questionnaire. Lastly, your experimenter will answer any questions you may have. Please sign below to indicate you have read and understand the instructions.

____

Volunteer Signature

EXPERIMENTAL INSTRUCTIONS

The purpose of this experiment is to collect information regarding navigation in a computerized database. You are asked to complete one session taking no more than two hours.

First, you will be given an introduction to the database. This database contains information about the states in three imaginary countries: Macaye, Kinsela, and Strathmore. Each country is composed of four provinces which are in turn composed of four states. Every state is represented by an "entry." Each entry is made up of three components: *text, table,* and *figure.* The *text* component gives information on the people and culture of the state. The *table* contains information about the state's economics and commerce. Lastly, the *figure* provides data on weather and the land. You are interested in vacationing in one of the states but you have not decided which one. You have come to the travel agent with a list of 65 questions. You are given access to the database to answer the questions sequentially.

Your experimenter will explain how to access text, table, and figure components. You will be given 12 sample questions to answer to give you practice with the database. Feel free to ask your experimenter any questions that you wish at this time.

You will then be asked to complete the remaining 53 multiple choice questions pertaining to information that can be found within the database. You may ask questions at any time but the experimenter reserves the right to delay some detailed explanations until the conclusion of both experimental sessions.

Try to answer the questions as quickly and accurately as possible. They are not trick questions! After you have completed the 65 questions, you will be asked to complete a brief questionnaire. Lastly, your experimenter will answer any questions you may have. Please sign below to indicate you have read and understand the instructions.

____

Volunteer Signature

APPENDIX D

Questionnaires

DATABASE NAVIGATION QUESTIONNAIRE

Please circle the number that corresponds to your experience or feelings about using the database:

1. How often do you use computers?

   1 2 3 4 5 6 7
   never ... daily

2. How easy or difficult was it to navigate through the database?

   1 2 3 4 5 6 7
   difficult ... easy

3. How efficient was the TABLE OF CONTENTS for finding information?

   1 2 3 4 5 6 7
   inefficient ... efficient

4. How efficient was GO BACK for finding information?

   1 2 3 4 5 6 7
   inefficient ... efficient

5. How easy or difficult was it to use the GO BACK function?

   1 2 3 4 5 6 7
   difficult ... easy

6. How well did the term GO BACK (or GO BACK and GO FORWARD) relate to its function?

   1 2 3 4 5 6 7
   unrelated ... related well

7. What is the maximum number of **states** you would go back using GO BACK rather than the table of contents?

   1 2 3 4 5 6 7

8. Would you prefer GO BACK to backtrack by state (i.e. Lytton to Trenton) or by components of states (i.e. Lytton text viewer to Trenton table viewer)?

   - ❑ state
   - ❑ component

9. Would you like GO BACK to provide a history list so that you may directly select a component that you viewed previously or do you prefer GO BACK to take you step by step back through the screens you have viewed?
   - ☐ provide a history list
   - ☐ go directly back

10. Do you have any suggestions to improve navigation and backtracking in this database?

Thank you for participating in this study!!

DATABASE NAVIGATION QUESTIONNAIRE

Please circle the number that corresponds to your experience or feelings about using the database:

1. How often do you use computers?

1	2	3	4	5	6	7
never						daily

2. How easy or difficult was it to navigate through the database?

1	2	3	4	5	6	7
difficult						easy

3. How efficient was the TABLE OF CONTENTS for finding information?

1	2	3	4	5	6	7
inefficient						efficient

4. Do you have any suggestions to improve navigation and backtracking in this database?

Thank you for participating in this study!!

APPENDIX E

Task Questions

Question Number	W1 (Anchor for W4)
Question	Use the Table of Contents to find what percentage of fresh water fish in Bluefield are perch.
Answer A	40%
Answer B	18%
Answer C	17%
Answer D	25%
Correct Response	B
Component	Bluefield Figure

Question Number	W2
Question	Use the Table of Contents to find how many species of marine life are at the National Aquarium in Tillson.
Answer A	30
Answer B	100
Answer C	1000
Answer D	2000
Correct Response	D
Component	Tillson Text

Question Number	W3
Question	Use the Table of Contents to find how many liver transplants are performed at St. Luke's Hospital in Sassinger.
Answer A	18
Answer B	16
Answer C	8
Answer D	2
Correct Response	A
Component	Sassinger Table

Question Number	W4 (Go Back Warm-Up 1)
Question	Use Go Back to find what percentage of salt water fish in Bluefield are flounder.
Answer A	36%
Answer B	29%
Answer C	11%
Answer D	24%
Correct Response	C
Component	Bluefield Figure

Question Number	W5 (Anchor for W7)
Question	Use the Table of Contents to find the average number of cats in the Costello animal shelter in 1989.
Answer A	300
Answer B	250
Answer C	350
Answer D	200
Correct Response	A
Component	Costello Figure

Question Number	W6
Question	Use the Table of Contents to find the number of non-resident fishing licenses sold in July of 1988 in Flint.
Answer A	3000
Answer B	4000
Answer C	5000
Answer D	6000
Correct Response	D
Component	Flint Figure

Question Number	W7 (Go Back Warm-Up 2)
Question	Use Go Back to find the average number of dogs in the Costello Animal Shelter in 1987.
Answer A	150
Answer B	200
Answer C	250
Answer D	300
Correct Response	B
Component	Costello Figure

Question Number	W8 (Anchor for W12)
Question	Use the Table of Contents to find the toll fee to travel from Conway to Seneca in Athens.
Answer A	$1.40
Answer B	$2.25
Answer C	$2.60
Answer D	$3.10
Correct Response	A
Component	Athens Table

Question Number	W9
Question	Use the Table of Contents to find how many major river systems are in Sutton.
Answer A	14
Answer B	32
Answer C	100
Answer D	150
Correct Response	A
Component	Sutton Text

Question Number	W10
Question	Use the Table of Contents to find the mosquito concentration of the city of Florence in Vaughn.
Answer A	0 to 1000
Answer B	1000 to 2000
Answer C	2000 to 3000
Answer D	3000 to 4000
Correct Response	B
Component	Vaughn Figure

Question Number	W11
Question	Use the Table of Contents to find the nationality of the Dunkards in Mitchell.
Answer A	English
Answer B	Italian
Answer C	French
Answer D	German
Correct Response	D
Component	Mitchell Text

Question Number	W12 (Choice Warm-Up 1)
Question	Find the toll fee to travel from Fairmont to Wallace in Athens.
Answer A	$0.75
Answer B	$1.75
Answer C	$2.75
Answer D	$3.75
Correct Response	C
Component	Athens Table

Question Number	5 (Anchor for 7)
Question	Use the Table of Contents to find the proportion of days that were rainy in Thorson in the month of June.
Answer A	30%
Answer B	40%
Answer C	20%
Answer D	50%
Correct Response	A
Component	Thorson Figure

Question Number	6
Question	Use the Table of Contents to find how many feet above normal Lake Maria rose in March of 1990 in Ridgewood.
Answer A	4
Answer B	3
Answer C	2
Answer D	1
Correct Response	B
Component	Ridgewood Figure

Question Number	7 (Choice 1)
Question	Find the proportion of days that were sunny in Thorson in the month of July.
Answer A	20%
Answer B	30%
Answer C	40%
Answer D	50%
Correct Response	C
Component	Thorson Figure

Question Number	8
Question	Use the Table of Contents to find what position Susan Betts from Mitchell held.
Answer A	Chief Justice
Answer B	Secretary of the Treasury
Answer C	Speaker of the House
Answer D	Ambassador
Correct Response	A
Component	Mitchell Table

Question Number	9 (Anchor for 13)
Question	Use the Table of Contents to find which city in Kaufmann has full access to roads I-50, Route 460, and Rt. 125.
Answer A	Clemens
Answer B	Fontano
Answer C	Imperial
Answer D	Lakebend
Correct Response	D
Component	Kaufmann Table

Question Number	10
Question	Use the Table of Contents to find how many textile plants are now in Rawlins.
Answer A	100
Answer B	200
Answer C	600
Answer D	800
Correct Response	C
Component	Rawlins Text

Question Number	11
Question	Use the Table of Contents to find what city in Northrup the Air and Space Museum is located.
Answer A	Topeka
Answer B	Einstein
Answer C	Venitia
Answer D	Longwood
Correct Response	A
Component	Northrup Text

Question Number	12
Question	Use the Table of Contents to find the elevation of the Delta Ranger Station on Mt. Needmore in Canterbury.
Answer A	300
Answer B	400
Answer C	500
Answer D	600
Correct Response	B
Component	Canterbury Figure

Question Number	13 (Go Back 2)
Question	Use Go Back to find the highway that the city of Eagle Stone has full acess to in Kaufmann.
Answer A	I-50
Answer B	Rt. 460
Answer C	Rt. 125
Answer D	.
Correct Response	C
Component	Kaufmann Table

Question Number	14 (Anchor for 18)
Question	Use the Table of Contents to find which county in Flint requires only a car safety inspection every 12 months and NO emission inspection.
Answer A	Sanilac
Answer B	Waterloo
Answer C	Wilson
Answer D	Yazzir
Correct Response	D
Component	Flint Table

Question Number	15
Question	Use the Table of Contents to find how much precipitation Foster actually received in July of 1990.
Answer A	4 inches
Answer B	3 inches
Answer C	2 inches
Answer D	1 inch
Correct Response	C
Component	Foster Figure

Question Number	16
Question	Use the Table of Contents to find what type of food the Red Sea Restaurant in Canterbury serves.
Answer A	Ethiopian
Answer B	German
Answer C	Cuban
Answer D	Indian
Correct Response	A
Component	Canterbury Text

Question Number	17
Question	Use the Table of Contents to find what proportion of Tillson citizens in 1981 were Chinese.
Answer A	12%
Answer B	15%
Answer C	16%
Answer D	23%
Correct Response	C
Component	Tillson Table

Question Number	18 (Choice 2)
Question	Find which of the following counties requires an emmission inspection in Flint:
Answer A	Harrisville
Answer B	Sanilac
Answer C	Waterloo
Answer D	Yazzir
Correct Response	C
Component	Flint Table

Question Number	19
Question	Use the Table of Contents to find how much the Governor of Canterbury earns.
Answer A	$124,000
Answer B	$118,000
Answer C	$98,000
Answer D	$78,000
Correct Response	A
Component	Canterbury Table

Question Number	20 (Anchor for 23)
Question	Use the Table of Contents to find what percentage of undergraduates from Victoria College in Watkins find jobs or are "placed" within 6 months
Answer A	29%
Answer B	52%
Answer C	65%
Answer D	81%
Correct Response	B
Component	Watkins Table

Question Number	21
Question	Use the Table of Contents to find what city the largest Roman Catholic Church is near in Athens.
Answer A	Dunkirk
Answer B	Palestine
Answer C	Briard
Answer D	Waterdam
Correct Response	A
Component	Athens Text
Question Number	22
Question	Use the Table of Contents to find how many tons of materials were recycled in Fishwick in 1970.
Answer A	1.8 million
Answer B	2.2 million
Answer C	3.2 million
Answer D	4.5 million
Correct Response	A
Component	Fishwick Figure
Question Number	23 (Choice 3)
Question	Find the percentage of undergraduates from Middlebury College in Watkins who were placed within 12 months.
Answer A	12%
Answer B	38%
Answer C	42%
Answer D	68%
Correct Response	D
Component	Watkins Table
Question Number	24
Question	Use the Table of Contents to find the number of coniferous trees per acre that Sutton had in 1970.
Answer A	50
Answer B	45
Answer C	40
Answer D	35
Correct Response	B
Component	Sutton Figure

Question Number	25
Question	Use the Table of Contents to find the number of police that Ablemarle County in Theodore has.
Answer A	30
Answer B	25
Answer C	24
Answer D	19
Correct Response	A
Component	Theodore Table

Question Number	26 (Anchor for 32)
Question	Use the Table of Contents to find how many species of plants may be found at Gilmore's theme park, the Jungle Book.
Answer A	2000
Answer B	3000
Answer C	4000
Answer D	5000
Correct Response	C
Component	Gilmore Text

Question Number	27
Question	Use the Table of Contents to find, on average, how many days of rain the city of Sayer in Brittany has each year.
Answer A	0-50
Answer B	51-100
Answer C	101-150
Answer D	151-200
Correct Response	D
Component	Brittany Figure

Question Number	28
Question	Use the Table of Contents to find the percentage of days in July of 1990 that had a high in the 70°s in Sandstrom.
Answer A	28%
Answer B	32%
Answer C	40%
Answer D	54%
Correct Response	B
Component	Sandstrom Figure

Question Number	29
Question	Use the Table of Contents to find how many dogs were registered in Hawthorne in 1975.
Answer A	30
Answer B	40
Answer C	50
Answer D	60
Correct Response	A
Component	Hawthorne Figure

Question Number	30
Question	Use the Table of Contents to find the length of the largest known natural rock arch in the world that is located in Amhearst.
Answer A	121 feet
Answer B	155 feet
Answer C	198 feet
Answer D	291 feet
Correct Response	D
Component	Amhearst Text

Question Number	31
Question	Use the Table of Contents to find the percentage of ticks carrying Lyme Disease in Largo County in Braxton.
Answer A	5%
Answer B	10%
Answer C	15%
Answer D	20%
Correct Response	B
Component	Braxton Figure

Question Number	32 (Go Back 3)
Question	Use Go Back to find how many African animals may be found in Gilmore's theme park, the Jungle Book.
Answer A	1000
Answer B	2000
Answer C	3000
Answer D	4000
Correct Response	A
Component	Gilmore Text

Question Number	33
Question	Use the Table of Contents to find how many bars and clubs there are on Paradise Island in Radford.
Answer A	21
Answer B	23
Answer C	25
Answer D	27
Correct Response	B
Component	Radford Text
Question Number	34 (Anchor for 41)
Question	Use the Table of Contents to find the average temperature in Armstrong in October at Noon.
Answer A	80°
Answer B	70°
Answer C	60°
Answer D	50°
Correct Response	C
Component	Armstrong Figure
Question Number	35
Question	Use the Table of Contents to find the prize money offered in the Fairchance Stakes horse race in Fairchance.
Answer A	$1 million
Answer B	$2.2 million
Answer C	$3.1 million
Answer D	$3.6 million
Correct Response	B
Component	Fairchance Text
Question Number	36
Question	Use the Table of Contents to find the amount of land sold for industrial and residential use in Mitchell in 1986.
Answer A	8,000 acres
Answer B	10,000 acres
Answer C	12,000 acres
Answer D	14,000 acres
Correct Response	C
Component	Mitchell Figure

Question Number	37
Question	Use the Table of Contents to find what type of program the radio station WHIQ offers in Appleton.
Answer A	Rock music
Answer B	Country music
Answer C	Talk Radio
Answer D	National Public Radio
Correct Response	D
Component	Appleton Table

Question Number	38
Question	Use the Table of Contents to find how many people participate in the Thanksgiving parade in Pamplin.
Answer A	3000
Answer B	1200
Answer C	900
Answer D	750
Correct Response	A
Component	Pamplin Text

Question Number	39
Question	Use the Table of Contents to find how many tons of materials were in Nottingham landfills in 1975.
Answer A	2,000,000
Answer B	3,000,000
Answer C	5,000,000
Answer D	8,500,000
Correct Response	C
Component	Nottingham Figure

Question Number	40
Question	Use the Table of Contents to find the average number of children per computer in 1990 for Sutton students in the 3rd grade.
Answer A	30
Answer B	25
Answer C	20
Answer D	18
Correct Response	B
Component	Sutton Table

Question Number	41 (Go Back 4)
Question	Use Go Back to find the average temperature in Armstrong in July at 6:00 am.
Answer A	30°
Answer B	40°
Answer C	50°
Answer D	60°
Correct Response	D
Component	Armstrong Figure

Question Number	42 (Anchor for 44)
Question	Use the Table of Contents to find the land area of Northrup.
Answer A	7,533 square miles
Answer B	8,523 square miles
Answer C	8,743 square miles
Answer D	9,124 square miles
Correct Response	B
Component	Northrup Table

Question Number	43
Question	Use the Table of Contents to find the depth of the snow base on Big Bear Mountain in Grafton in December.
Answer A	40 inches
Answer B	35 inches
Answer C	30 inches
Answer D	25 inches
Correct Response	D
Component	Grafton Figure

Question Number	44 (Go Back 5)
Question	Use Go Back to find the Capital of Northrup.
Answer A	Fairfax
Answer B	Georgetown
Answer C	Alexandria
Answer D	McLean
Correct Response	B
Component	Northrup Table

Question Number	45
Question	Use the Table of Contents to find how many buses are used in Thorson.
Answer A	800,000
Answer B	1500
Answer C	100
Answer D	75
Correct Response	B
Component	Thorson Table

Question Number	46 (Anchor for 53)
Question	Use the Table of Contents to find how many rooms are at the Homestead Resort in Costello.
Answer A	620
Answer B	520
Answer C	420
Answer D	320
Correct Response	A
Component	Costello Text

Question Number	47
Question	Use the Table of Contents to find which month has the largest amount of mold in Woodward's air.
Answer A	March
Answer B	April
Answer C	May
Answer D	June
Correct Response	C
Component	Woodward Figure

Question Number	48
Question	Use the Table of Contents to find how many bucks were shot by hunters in Greenwalt in 1984.
Answer A	160
Answer B	140
Answer C	120
Answer D	100
Correct Response	D
Component	Greenwalt Figure

Question Number	49
Question	Use the Table of Contents to find the staff to visitor ratio at the Cool Springs Health Resort in Foster.
Answer A	5:1
Answer B	4:1
Answer C	2:1
Answer D	1:1
Correct Response	C
Component	Foster Text

Question Number	50
Question	Use the Table of Contents to find the average temperature of the water at Lake Anna in Sassinger in November.
Answer A	60°
Answer B	50°
Answer C	40°
Answer D	30°
Correct Response	B
Component	Sassinger Figure

Question Number	51
Question	Use the Table of Contents to find the total number of poisonous snake bites reported in Ironto in 1975.
Answer A	30
Answer B	25
Answer C	20
Answer D	15
Correct Response	B
Component	Ironto Figure

Question Number	52
Question	Use the Table of Contents to find how many stories the Popco Tower in Partridge is.
Answer A	91 stories
Answer B	97 stories
Answer C	114 stories
Answer D	115 stories
Correct Response	D
Component	Partridge Text

Question Number	53 (Choice 4)
Question	Find how many pools are at the Homestead Resort in Costello.
Answer A	1
Answer B	2
Answer C	3
Answer D	4
Correct Response	C
Component	Costello Text

APPENDIX F

Descriptive Statistics

TABLE 1

Mean Time (in seconds) for each Dependent Measure Within and Across Trials for the Go Back Trials For Each Group

	Trial/Task Length					
Group / Time Measure	***Trial 1*** ***2***	***Trial 2*** ***5***	***Trial 3*** ***9***	***Trial 4*** ***1 1***	***Trial 5*** ***3***	***Mean***
Component/History List						
Time to First Action	6.50	7.87	5.12	6.50	4.37	6.07
Navigating	13.62	14.37	19.25	30.75	18.50	19.30
Answering	8.37	12.50	7.12	16.75	4.00	9.75
Entry/History List						
Time to First Action	5.25	5.87	5.50	4.75	4.50	5.17
Navigating	6.50	12.87	9.37	11.50	16.50	11.35
Answering	10.00	12.00	7.87	10.87	3.37	8.82
Component/No History List						
Time to First Action	8.12	6.62	6.50	5.12	5.12	6.30
Navigating	6.12	12.00	23.50	24.62	5.25	14.30
Answering	11.87	10.87	8.50	14.75	4.87	10.17
Entry/No History List						
Time to First Action	6.37	6.50	6.50	7.87	8.37	7.12
Navigating	7.50	14.87	11.25	16.75	23.37	14.75
Answering	11.12	11.12	7.62	16.62	3.50	10.00
Control						
Time to First Action	6.75	4.50	5.25	2.62	2.75	4.37
Navigating	26.37	31.75	24.88	39.63	43.88	33.30
Answering	10.50	7.25	2.37	10.87	5.12	7.22

TABLE 2

Mean, Standard Deviation, and Median Navigation Time (in seconds) for each Subject Across Go Back Trials Within Each Group

Group/Measure	Subject 1	2	3	4	5	6	7	8	Mean
Component/No History List									
Mean	37.0	21.2	29.8	9.0	22.4	9.8	10.0	15.2	19.3
Standard Deviation	40.2	13.1	22.8	5.6	13.8	4.3	5.1	11.6	14.6
Median	21	18	25	7	20	10	10	11	15.3
Entry/History List									
Mean	15.2	10.8	13.2	10.0	14.2	7.8	11.6	8.0	11.4
Standard Deviation	8.4	8.0	5.9	7.6	4.7	3.8	4.1	3.5	5.8
Median	13	7	12	9	12	8	10	7	9.7
Component/No History List									
Mean	12.0	18.6	13.4	11.4	18.6	14.4	11.6	14.4	14.3
Standard Deviation	8.2	13.1	12.2	8.3	11.6	14.6	11.1	7.3	10.8
Median	9	16	7	7	20	9	6	15	11.1
Entry/No History List									
Mean	8.2	12.2	29.4	24.4	7.8	7.6	11.2	17.2	14.8
Standard Deviation	4.3	7.5	20.2	27.5	5.5	4.0	5.8	11.0	10.8
Median	9	15	16	16	7	9	13	19	13.0
Control									
Mean	25.0	47.2	37.2	24.6	24.4	49.4	21.0	37.6	33.3
Standard Deviation	11.3	9.6	22.0	11.2	11.5	4.7	6.6	19.0	12.0
Median	26	51	36	26	25	49	17	39	33.6

TABLE 3

Number of subjects (out of eight) using Go Back on each choice trial followed by average number of Go Back choices (out of four trials) made in each Group

GROUP	Choice Trial 1	2	3	4	Average Number out of Four Trials
Component - History	6	7	6	6	3.13
Entry - History	8	8	8	7	3.88
Component - No History	5	7	7	6	3.13
Entry - No History	4	5	4	3	2.00

TABLE 4

Mean Responses to Questionnaire on a Seven-Point Rating Scale by Group

GROUP	1	2	3	4	5	6	7
Component - History	4.88	6.00	5.50	6.13	6.00	5.38	5.88
Entry - History	4.25	6.38	3.75	6.38	7.00	6.00	6.00
Component - No History	4.50	6.00	5.38	5.50	6.63	5.75	4.88
Entry - No History	5.00	5.88	3.75	6.00	6.50	6.38	5.00
Control	4.00	5.63	5.38	n/a	n/a	n/a	n/a

APPENDIX G

Results of Statistical Analysis

TABLE 1

One Factor ANOVA on Mean Time To First Action

Source	*df*	*SS*	*MS*	*F*	*p*
Group	4	36.016	9.004	2.37	0.071
Subj (Group)	35	133.100	3.802		

TABLE 2

One Factor ANOVA on Mean Navigating Time

Source	*df*	*SS*	*MS*	*F*	*p*
Group	4	2419.640	604.910	9.69	0.0001
Subj (Group)	35	2183.960	62.399		

TABLE 3

One Factor ANOVA on Mean Answering Time

Source	*df*	*SS*	*MS*	*F*	*p*
Group	4	10.690	2.672	0.242	0.9129
Subj (Group)	35	386.110	11.032		

TABLE 4

One Factor ANOVA on Median Navigation Time

Source	*df*	*SS*	*MS*	*F*	*p*
Group	4	3052.150	763.038	15.79	0.0001
Subj (Group)	35	1691.750	48.336		

TABLE 5

One Factor ANOVA on Standard Deviation of Navigation Time

Source	*df*	*SS*	*MS*	*F*	*p*
Group	4	322.947	80.736	1.52	0.217
Subj (Group)	35	1858.294	53.094		

TABLE 6

Significant Differences Between Combinations of Trial and Group. Numbers Across Top and Down Left Indicate the Trial Number and X's Signify a Significant Difference as Determined by the LSD Test.

		COMP/ HIST					ENTRY/ HIST					COMP/ NO					ENTRY/ NO					CONTROL				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
CONTROL	5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X		
	4	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X		
	3						X		X	X		X				X	X		X							
	2	X				X	X	X	X	X	X	X	X			X	X	X	X	X						
	1						X	X	X	X		X	X			X	X		X							
ENTRY/ NO	5						X		X			X				X										
	4				X																					
	3				X										X											
	2				X																					
	1				X									X												
COMP/ NO	5			X	X	X								X	X											
	4						X		X	X		X														
	3						X		X			X														
	2				X																					
	1			X	X																					
ENTRY/ HIST	5				X																					
	4				X																					
	3				X																					
	2				X																					
	1			X	X																					
COMP/ HIST	5																									
	4	X	X																							
	3																									
	2																									
	1																									

TABLE 7

Two Factor ANOVA on Mean Navigating Time

Source	df	SS	MS	F	p
Level	1	112.500	112.500	2.38	0.134
History	1	5.120	5.120	0.11	0.744
Level x History	1	141.120	141.120	2.99	0.095
Subj (Level History)	28	1322.760	47.241		

TABLE 8

Two Factor ANOVA on Median Navigation Time

Source	df	SS	MS	F	p
Level	1	26.281	26.281	1.13	0.297
History	1	1.531	1.531	0.07	0.800
Level x History	1	108.781	108.781	4.67	0.039
Subj (Level History)	28	651.875	23.281		

TABLE 9

Two Factor ANOVA on Choice Frequency

Source	***df***	***S S***	***M S***	***F***	***p***
Level	1	0.281	0.281	0.16	0.696
History	1	7.031	7.031	3.89	0.058
Level x History	1	7.031	7.031	3.89	0.058
Subj (Level History)	28	50.625	1.808		

TABLE 10

One Factor ANOVA on Reported Computer Experience

Source	***df***	***S S***	***M S***	***F***	***p***
Group	4	5.600	1.400	0.40	0.807
Subj (Group)	35	122.375	3.496		

VITA

Cortney G. Vargo

509 B Sunridge Drive
Blacksburg, VA 24060
(703) 951-3637
CORTV@VTVM1.vt.cc.edu

EDUCATION

Master of Science, human factors engineering option of Industrial and Systems Engineering, May 1992. Virginia Polytechnic Institute and State University, Blacksburg, VA. QCA: 3.95/4.00.

Thesis: "An Evaluation of Computer-Supported Backtracking in a Hierarchical Database."

Bachelor of Science degree *magna cum laude*, Industrial Engineering and Operations Research, May, 1989. Virginia Polytechnic Institute and State University, Blacksburg, VA.

PROFESSIONAL EXPERIENCE

TEACHING

Teaching Assistant, Engineering Economy (Fall 1989, Spring 1990, Spring 1991) and Methods and Measurement Engineering (Summer, 1990), *Virginia Polytechnic Institute and State University.* Responsible for all areas of classroom management: lecturing, grading, test development, office hours, help sessions. Received an excellent student review.

Horseback Riding Instructor, *Hillside Equestrian Center*, Pittsburgh, PA (Summers 1987, 1989). Taught equitation and stable management to 25 students.

RESEARCH

Student Research Associate, *Air Force Office of Scientific Research, Wright-Patterson Air Force Base, Armstrong Laboratories, Human Engineering Division*, Dayton, OH (Summer, 1991). Coordinated and completed a computer interface usability study examining the effects of two backtracking algorithms and two interface types on navigating time in an interactive hierarchical computerized database. Responsible for all phases of the study: experimental design, software development, running subjects, and analyzing data. Results will be incorporated in a large-scale human engineering database currently under development by the Air Force.

Research Assistant, *Virginia Polytechnic Institute and State University.* (Fall, 1990). Worked with Dr. Robert C. Williges to produce an informational videotape about teleconferencing. Wrote a manual for developing high quality, low cost videotapes for remote lecturing.

PROFESSIONAL EXPERIENCE (cont.)

Student Assistant, *Management Systems Laboratories*, Blacksburg, VA (Spring 1989, Summer 1990). Wrote sections of an emergency management plan for the Department of Energy. Coordinated the desktop publishing and developed all graphical support. Did literature searches and reviews.

Laboratory Assistant, *Mobay Chemical Corporation*, *Polyurethane Research Division*, Pittsburgh, PA (Summer 1986). Maintained lab, conducted hydrolysis experiments, collected and recorded data.

ENGINEERING

Consultant, *Titan Equipment Corporation*, Pittsburgh, PA (Summer 1989). Created office layouts, implemented information flow improvements, and developed computerized product tracking and forecasting methods.

Industrial Engineer, *Litton Poly-Scientific*, Blacksburg, VA (Spring to Fall 1988). Developed methods improvements to decrease scrap and rework in job shop production.

Summer Analyst, *Mobay Chemical Corporation, Management Development Deparment*, Pittsburgh, PA (Summer 1987). Implemented scheduling software for managerial training programs.

ACTIVITIES

- Secretary, *Graduate Student Assembly*, 1990-1992.
- Vice President, *Graduate Student Assembly*, 1992-1993.
- Secretary, *Engineering Graduate Student Committee*, 1989-1990.
- Graduate Representative to the *University Council*, 1990-1992.
- Graduate Representative to the *Library Committee*, 1991-1992.
- *Kappa Delta Sorority*, Scholarship Chairman, 1987-1989.
- *United States Pony Clubs*, life member, instructor and judge, 1981- present.

PROFESSIONAL SOCIETIES

- *Human Factors Society*, 1989- present
- *Association of Computing Machinery* (ACM), 1990- present
- *Special Interest Group (ACM) for Computer-Human Interaction*, 1990-present
- *Tau Beta Pi* National Engineering Honor Society, 1988-present
- *Omicron Delta Kappa* National Leadership Honor Society, 1992-present

Cortney Vargo

Cortney G. Vargo

Birthdate: March 22, 1967