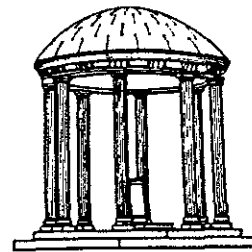# Navigational Techniques to Improve the Display of Large Two-Dimensional Spaces

TR89-042 REVISED

October, 1989

David V. Beard
John Q. Walker II

The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175

Navigational Techniques to Improve the Display of Large Two-Dimensional Spaces

April 21, 1989

David V. Beard, Ph.D.
John Q. Walker II

The University of North Carolina
Departments of Radiology and Computer Science
Chapel Hill, North Carolina   27514

IBM Corporation
Communication Systems
Research Triangle Park, North Carolina   27709

Abstract

Often the components of a problem can be arrayed on a two-dimensional information space (for example, as an abstract tree or hypertext) far too large to fit onto a computer display. With current navigation techniques it is often difficult for users to keep track of their location or to move rapidly to remote locations in the space.

We implemented two similar direct-manipulation techniques that both use a *map window*--a miniature of the entire information space--with a wire-frame box to aid users in remembering their location. The first technique allows the user to rapidly *roam* over the information space by moving the location of the wire-frame box. The second allows for *zooming* as well as roaming.

A controlled experiment compared the above techniques to scroll bars for determining whether a target word was in a large balanced binary tree of words. The experiment also examined the merit of the map window.

Map windows significantly improved user performance, and the roam and zoom techniques were significantly faster than the scroll bars. Our observation of subjects and their verbal protocol indicated that a paging feature on the scroll bars--allowing rapid movement of the screen a page in any direction--was advantageous.

Keywords

Contents

Introduction

Often the components of a problem can be arrayed on a two-dimensional information space. Sometimes the information has a visual analog that makes the meaning of the information space clear to the user; a city map or a cat-scan medical image, for example. At other times, information can be represented as a two-dimensional logical graph consisting of arcs and nodes; for example, structure charts, data flow diagrams, PERT charts, hypertext graphs, and spatial databases.

If the two-dimensional information space fits completely onto a display screen, there is no navigation problem. Users can point to any location using one of the computer's pointing devices. They are never lost because they can always see the complete information space. But what if the information space is considerably larger than the display screen? What if the user wishes to navigate and examine a graph consisting of 10,000 nodes and arcs?

Research Description

While moving a cursor on a single screen is relatively well-understood, navigating in a two-dimensional information space consisting of hundreds of screen-size pages has not been so well studied. What navigational techniques allow a user to locate and view both nearby and remote regions in a large, two-dimensional information space?

We implemented two similar direct-manipulation techniques that both use a map window--a miniature of the entire information space--to aid the user in remembering his location. The first technique allows the user to rapidly *roam* over the information space. The second allows for *zooming* into the space as well as roaming. The zoom technique is especially well-suited for hardware implementations that can support such a feature.

A controlled subject experiment compared the above techniques to scroll bars for determining whether a target word was in a large balanced binary tree of words. Both horizontal and vertical scroll bars were used to allow two-dimensional movement. This three-by-two experiment additionally examined the merit of map windows.

Map windows significantly improved user performance, and the roam and zoom techniques were significantly faster than scroll bars. Our observation of subjects and their verbal protocol indicated that a paging feature on the scroll bars--allowing rapid movement of the screen a page in any direction--was advantageous.

The following section contains a summary of previous work in this area. Next, the two navigational techniques are described. The controlled experiment is then described along with results, conclusions, and recommendations.

Previous Work

Many computer applications have been developed using two-dimensional graphical information spaces. The following section presents an overview of the shared problems encountered with these systems, and the various solutions developed by their designers.

Information Systems Using Two-dimensional Spatial Mental Models

The information spaces discussed in this paper are "mental models" analogous to physical planes or pieces of paper, presenting the user with the information in spatial form. Therefore, the organization of the information is understood prior to system use. The rules of physical movement also apply, *e.g.*, left, right, zoom-in, and zoom-out.

A mental model is 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. Spatial mental models may not be the ubiquitous solution to all information search problems and spatial searching may not always be as powerful as simple linguistic tags (Dumais and Jones, 1985). However, as long as a mental model is well understood--for example, it is analogous to some mental concept the user already understands--and as long as the system does not move outside the ideas embraced by a model (Young, 1981; Douglas and Moran, 1983), a mental model is a powerful tool to reduce learning time and improve overall performance (Rumelhart and Norman, 1981; Halasz and Moran, 1982; Foss, Rosson, and Smith, 1982).

Hypertext

An increasingly common use of two-dimensional information spaces is *hypertext*. Hypertext (Bush, 1945; Englebart, 1963; Nelson, 1965; Englebart and English, 1968) commonly refers to a collection of labeled nodes connected by labeled links, where each node contains a useful chunk of information such as text, graphics, or sound.[1] Conklin (1987) and the Hypertext '87 conference proceedings (1987) provide overviews of current hypertext systems and their features.

Unlike other information systems, a predominant method for finding nodes and links in hypertext is a human visual search. While several hypertext systems allow search by following links between nodes, other hypertext systems encourage a mental model of a two-dimensional information space upon which all the nodes and links are arrayed. Their designers propose that users need

---

[1] Formally, it is a restriction of the graphic-theoretic semantic net (Quillian, 1968; Griffith, 1982), trading conceptual simplicity for decreased expressive power.

not construct a new mental model of the node layout, but rather can use their existing mental models of physical planes to locate needed items. This is done by spatial references as well as simply following links (*cf.* Delisle and Schwartz, 1986; Halasz, Moran, and Trigg, 1987). However, visual searching in hypertext can be difficult, and users often become lost and make unsuccessful search decisions (Mantei, 1982). A number of hypertext systems include "browsers" to ease their navigational difficulties.

Other Spatial Information Systems

Two-dimensional information spaces are also used to express and search other sorts of information besides hypertext graphs and trees. Geographic information systems are one example where the two-dimensional information space has a clear analogy to an object--a physical map--with which the user is familiar (*cf.* Frank, 1982). With geographic information systems, the user can locate information by moving through a two-dimensional map displayed on a computer display and pointing to some location. Information related to that location, such as the population density or natural resources, can then be displayed.

Other examples of information systems using a two-dimensional information space with a physical analog include radiology workstations and the UNC building walkthrough system. Radiologists typically view medical images of physical film displayed on a array of light boxes. When developing a computer workstation for radiologists, it makes sense to use a physical analogy of this situation and provide the same mental model to the user (Beard *et al.*, 1987; Beard and Symon, 1988). Brooks' building walkthrough system (1986) allows a user to navigate the floor plan of a building while a separate real-time display shows a three-dimensional view of what the user would see at that point in the building.

It is often advantageous to display information visually that is otherwise not considered spatial. The spatial information management system (Donelson, 1978) allows the user to "fly" through various information spaces. The spatial data management system (Herot, 1980) can be used to spatially locate logical information stored in a traditional database management system. Caplinger (1986) presents a summary of these and other graphic information systems. Another example is text editing. A text file can be thought of as a one-dimensional information space through which the user navigates to locate the desired line of text.

Navigating in a Large Two-dimensional Space

Two-dimensional information spaces such as hypertext often require *navigation*. Navigation, that is, viewing and manipulating the computer display to show another portion of the information space, is necessary when the display medium is too small or is of insufficient resolution to present the whole image. Navigation is thus a non-essential portion of a user's actual task (Brooks, 1987); it is a means to the user's end, not an end in itself. As a non-essential by-

product of computing, navigation should consume as few *cognitive cycles* as possible.

Navigation involves two elements:

1. A user thinks "I want to be at such-and-such a location." Everything from the moment of that thought until reaching the desired location is non-essential, and therefore cognitive overhead.

2. A user thinks "Where am I?" Everything from the moment of that question until its successful answer is also cognitive overhead. The cost in cognitive cycles of becoming truly lost is enormous.[2]

Since the information space is too large to be displayed in its entirety, navigation operates by suppressing some amount of detail. In a visual medium, this is done using the following methods, singly or in combination. First, *truncating*, where the information space is cropped so only a portion is seen. Second, *shrinking*, with elements below a given resolution being lost. And finally, *distorting*, where elements in the central focus are larger than elements at a greater distance from the center (for example, a fisheye lens).[3]

The truncating method is typically used in text editors. Only a small portion of the text is actually visible on the screen at a time. Users have to guess "where they are." Line numbers, scroll bar indicators (Kay, 1977), and other methods are used to try to minimize the navigation problems.

A navigation or *map* view, based on the shrinking method, is used to help people quickly learn their location and move about the information space. Maps help people keep track of their location (Baird, 1979). In thinking about maps to aid navigation, we considered how we use a large road map in a car. While the map usually covers everything we want to see, we typically focus on a smaller local region--and fold the map up to show just that region.

Map views provide a miniature of the entire information space that can be displayed simultaneously with a "you-are-here" indicator. Often this indicator is a wire-frame rectangle indicating a small portion of the information space which is then displayed in a larger *detail view* window. The map view concept is found in many existing spatial information systems, including systems described by Donelson (1978), Brooks (1986), Delisle and Schwartz (1986), Smith

---

[2] We have found a *virtual memory* analogy to help computer software designers--not necessarily users-- understand and communicate aspects of the navigation problem. As with virtual memory in a computer system, the portion of the information space actually being displayed can be thought of as being in a computer's main memory, while the remaining non-displayed portion of the information space is in secondary storage. Manipulating the system to display a new portion of the information space then is referred to as a *page fault*. The goal of the information-space system designer is to avoid *thrashing*; that is, to make sure that manipulating the system to display needed portions of the information space does not disrupt the user's real task with excessive cognitive overhead.

[3] Similar methods are used in logical media. For example, a table of contents represents the shrinking method-- elements below a certain resolution (*e.g.*, sub-sections of a book) are not visible.

et al. (1986), and Halasz *et al.* (1987). Variations of map views are often found in computer games, CAD/CAM and drafting packages, word processing packages such as Lotus *Manuscript*™, and VLSI tools such as *ICARUS* (Fairbairn and Rowson, 1978). A map view should either take up little valuable screen space or minimize the effort needed to show and hide it.

We are aware of one example based on the distorting method mentioned above. Furnas (1986) uses the fisheye concept to display the entire information space. The center portion of the display shows information at full resolution, while the image is increasingly compressed as the edges are approached.

Manipulating the Display

Users manipulate their location in large two-dimensional information spaces such as hypertext by using several common techniques. These techniques differ in two areas: 1) in the ease with which they allow diagonal or curvilinear movement (the problem on a classic *Etch-A-Sketch*, for example), and 2) in the ease in which they allow rapid access to remote regions of the space. Typically, users navigate within a spreadsheet, for example, with cursor keys and scroll bars to move horizontally or vertically among the cells (Hayes, 1985). Microsoft® *Windows* supplies horizontal and vertical scroll bars to move its viewport window over a large underlying, two-dimensional space (Microsoft, 1985). The Apple Macintosh® *MacDraw* allows two-dimensional navigation with a *hand* that grabs the space and scrolls it under the viewport.

We did not specifically study navigation using the cursor keys or the hand, since, while good for fine-focusing, we observed that they were unsuitable for movement to remote regions of the information space. Note, however, that with sufficient speed, some fine-focus access methods can replace coarse-focus ones. For example, if the scroll-down key on a text editor is sufficiently fast, the user may simply use it for all navigation, including long distances.

Other Issues for Navigating in Two-dimensional Information Spaces

Several other navigation issues need to be considered by the computer scientist designing a two-dimensional information system such as hypertext.

- With hypertext systems, the question arises as to whether the graph or tree should be static or dynamic, that is, reformatted each time the user manipulates the display. On the one hand, with a reformatted graph or tree, a larger portion of the information space can be displayed at the same resolution. On the other hand, the distorted shape of the information space may confuse the user, negating the entire reason behind the two-dimensional mental model in the first place.

- With many visually-oriented applications, the user may require simultaneous, detailed viewing of several portions of the information space. One example is the radiologist who must compare the size of a tumor last month with the cat-scan study from this month (Beard and Symon, 1988). Similarly, a writer may wish to look at parallel structures in the introduction and conclusion of an essay (Smith and Smith, 1987). It appears to be

advantageous for the "you-are-here" indicators from all the detail views to be displayed and possibly manipulated on the same map view.

- Several two-dimensional information systems navigate in information spaces so large that they use a recursive information space concept. Any detail view can become a map view, with a rectangle showing the location of still another detail view (Donelson, 1978). This approach greatly increases the richness of the information space with the possibly corresponding increase in mental complexity for the user.

The Two Navigational Techniques Studied

Objectives

The objective of these two navigational techniques is to allow easy access to various locations in a two-dimensional information space, including both near and far perpendicular and diagonal motion. These techniques are application-independent; they can be used for a general class of computer interaction problems that involve locating current position and moving through a two-dimensional information space considerably larger than the display screen. While their functions differ, the techniques share an underlying mental model and a considerable overlap in interaction style.
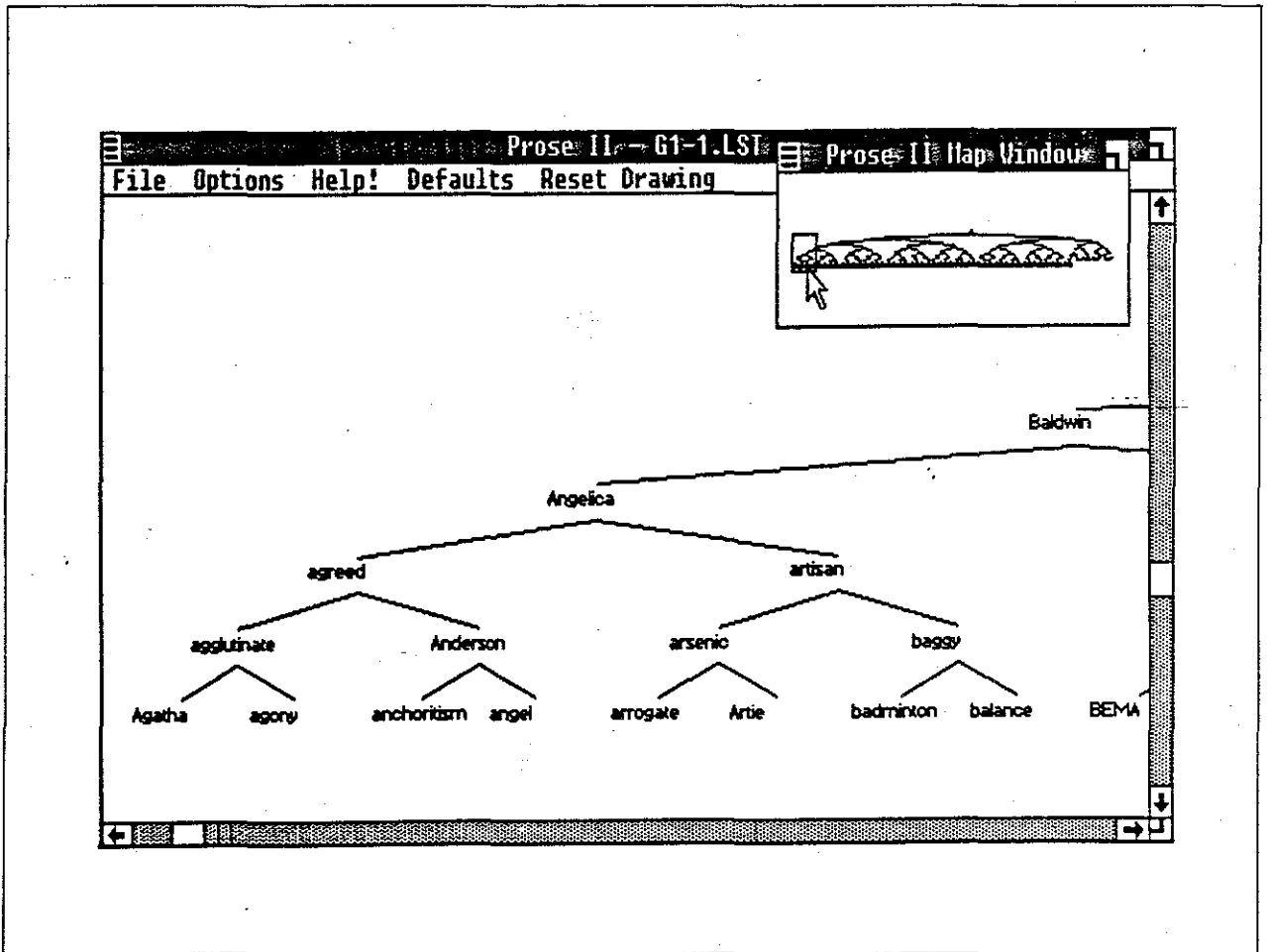
Figure 1. A typical display screen in our experiment.

Map Window

Figure 1 on page 7 illustrates the display for *Prose II*. The design for this tree-based structure editor is the result of work on previous hypertext systems and ideas by the Writing Research Group at the University of North Carolina (Smith *et al.*, 1985). In the upper right hand corner of the screen is a small window--the map window--showing in miniature the entire two-dimensional information space. A rectangle (called here the *wire-frame box*) inside the map window denotes the portion of the information space that is shown in the main display. This main display becomes a viewport into the larger information space.

The map window takes up screen space, but in return provides a clear notion of where the user--or more to the point, his current viewport--is within the information space. Further, it allows direct manipulation (Schneiderman, 1983; Schneiderman, 1987, Chap. 5) of the viewport within the space. For many applications, the trade-off of screen space for better navigation is worthwhile. If screen space is critical, the map window can be made to only appear when desired.

The contents of the map window used here were displayed in anisotropic map mode. The map window itself could be re-sized and re-shaped by a user. Its displayed contents would consequently change its size and aspect ratio as the map window changed.

Zoom Navigational Technique

To change the location or the magnification factor of the main display, the user moves the cursor--using a mouse--into the map window, presses the mouse button, drags the cursor down and to the right, and releases the button drawing a new wire-frame box in the map window. The original wire-frame box disappears. As in the previous case, the magnified contents of the new box are shown in the main display. If the new box is within the old, the user has zoomed in. This technique allows a user to zoom out, or roam around the screen. The user can also zoom into the space by drawing a wire-frame box within the main display itself using the right button on the mouse. We dedicated a mouse button to zooming in our implementation, giving us immediate access to the zoom feature, and more precise control over the magnification and size of the image in the main display.

Roam Navigational Technique

Both roaming and zooming within the information space may be useful in a single implementation; however, for many applications it may be either too slow for the given hardware-software combination, or inappropriate for the users' task. To this end, we have an alternate navigational technique that allows roam but not zoom. Like the zoom technique, a map window is provided and a wire-

frame box within the view denotes the location and size of the main display. However, with the roam technique the magnification is fixed at full resolution and the dimensions of the wire-frame cannot be changed. Instead of drawing a new wire-frame box, the user *grabs* the existing box with the mouse button, drags the box to its new location, and *releases* it. The main display then shows the region of the information space within the new location of the wire-frame box.

Experiment

Goal: The purpose of the study was to evaluate the usefulness of the roam and zoom techniques by comparing them to scroll bars (both horizontal and vertical), and to evaluate the effectiveness of map windows.

Hypotheses:

1. Roam and roam-and-zoom each will be faster than the scroll bars, both with and without the presence of the map windows.

2. Each of the techniques with the map window would be faster than without.

We believe these two hypotheses are predictable. A "back-of-the-envelope" time-motion calculation (such as one based on Fitts, 1954) tends to support the first hypothesis for rapid movement on the screen. Grabbing one object on the screen is likely to be much faster than grabbing and moving two. Further, the total distance needed for movement, and the different sizes of the grabbed objects would both tend to suggest slower times for scroll bars. The one exception would be if a preponderance of movements required only horizontal or only vertical movement. The map windows provide context. Roaming to unknown locations in the space should be faster because the subject only has to move to a location, not also find it.

Subjects: Six computer science graduate students were used as subjects in the experiment--three males and three females, with ages ranging from 23 to 26. The experience of these subjects ensured that mouse manipulation or the task of searching for words in the dictionary did not overwhelm the effect of the navigational techniques.

Apparatus: An IBM Personal Computer AT® with an IBM Enhanced Graphics Adapter and Display and a Microsoft mouse were used throughout this experiment. This monitor has a resolution of 640×350 pixels. The structure editor used in the experiment was written using the *Microsoft Windows Software Development Kit*. The zoom technique was implemented in software, without any special hardware assist.

The study was conducted in a well-lit office isolated from external sound. Subjects were comfortably seated in front of the computer and were allowed to adjust the location and angle of the monitor, and the location of the mouse. The experimenter sat behind and to the right of the subjects and used the computer's keyboard to control the experiment. Times were recorded manually using a stopwatch.
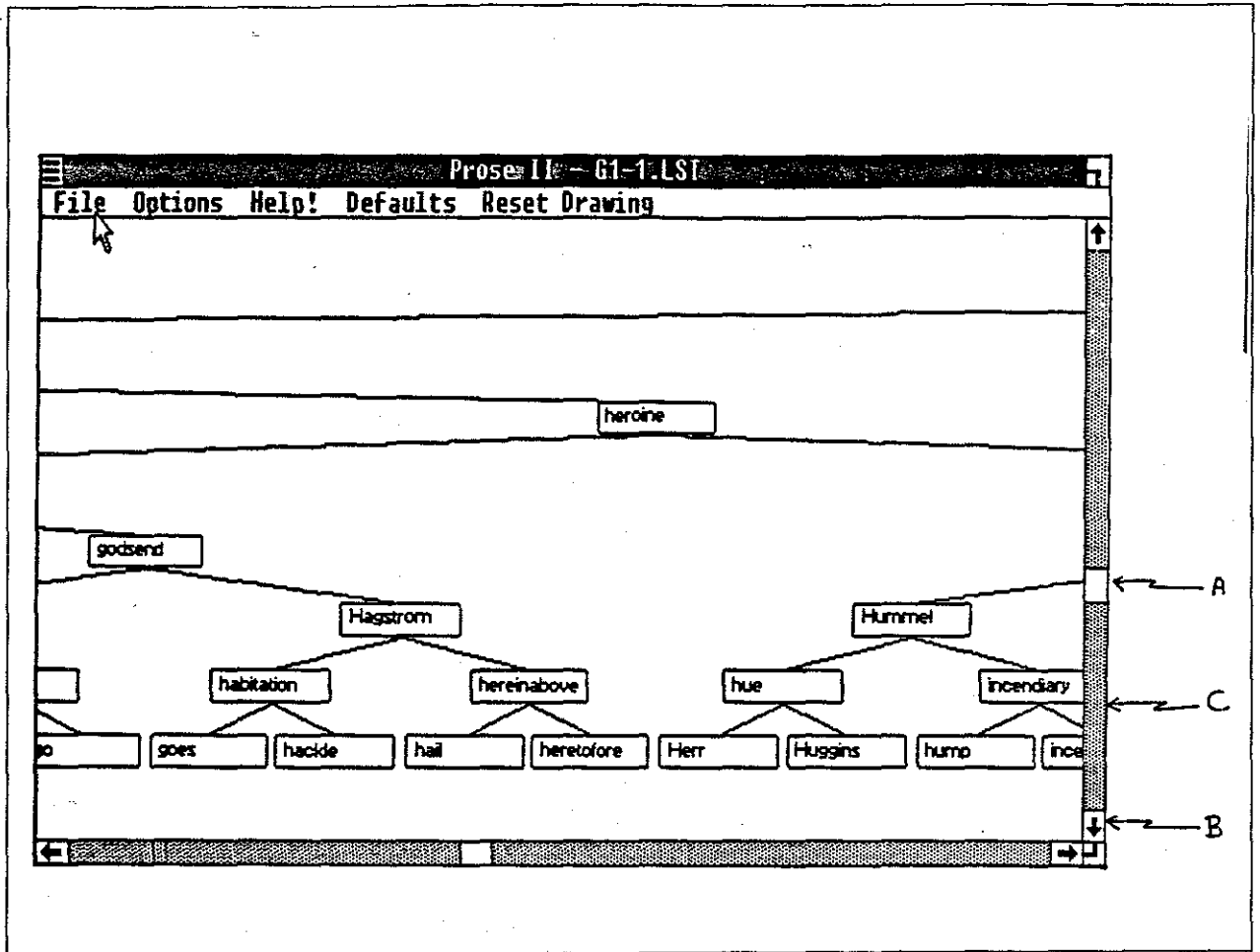
Figure 2. Scroll bar features, as implemented in the Prose II structure editor.

The scroll bars used in the experiment appear along the bottom and far right of a display window (see Figure 2). The location of the viewport in the underlying two-dimensional information space can be controlled using three techniques. First, the mouse can be used to grasp the scroll (A) and move it to a new location, changing the viewport to the corresponding location in the information space. Second, one of two small arrows (B) at either end of the bar can be clicked with the mouse. This causes the main display to scroll in the direction indicated by the arrow, for a distance equal to the size of a node in that dimension. Finally, if the region between the bar and the arrow is clicked (C), the bar moves one page (the size of the main display) in the direction of the arrow.

The three navigational techniques were each used with and without map windows to determine the significance of displaying, in miniature, the entire information space. However, both the roam and zoom techniques require the presence of the map window on which to operate. Therefore, when the map window was not desired, it was displayed empty, that is, without a miniature of the underlying information space.

As can be seen from Figure 2, the map window, while detailed enough to show the span of the binary tree, is unable to display the labels of the individual nodes. Thus, users cannot locate exact words using the map window; they can only move to a general region in the tree using their spatial and cognitive abilities.

Tasks: The effectiveness of a navigational technique may depend on the user's *cognitive resources* available (see Card, Moran, and Newell, 1983). To fully evaluate a technique, two different tasks were used: the first task was relatively complex, and the second relatively simple.

Task One required the subject to find a word in a balanced binary dictionary tree in which the labels of the nodes were dictionary words. Thus the subjects were navigating through the dictionary tree while doing the reasonably-complex cognitive task of work recognition and binary search.

Task Two required the subject quickly to move the viewport from displaying the apex node of a binary tree to displaying the rightmost leaf node--and back--five times. Thus, the subjects could fully concentrate on the navigational task.

Procedure: On being admitted to the office, the experimenters greeted the subjects, gave them a brief description of the experiment and its purpose, and obtained background data from them. They were then shown how to use the scroll bars, roam technique, and zoom technique in varying order. Each subject was trained and tested on Task One, followed by Task Two. Throughout the experiment, observations and verbal protocol were manually recorded. Subjects were instructed to "work as quickly as possible without making errors."

For each technique the following procedure was used. First, four training trials were given using the technique with the map window to familiarize the subject with the technique. Then, four training trials were given with or without the map window followed by a brief rest break with eyes closed (to avoid eye fatigue), followed by five timed trials. After a one minute break, four additional training trials were given, followed by an eyes-closed rest break and five timed trials.

An individual trial for Task One proceeded as follows: the subject was shown a 3x5 card on which the target word was written. After about four seconds, the subject was asked "Ready?" and then a verbal "Go" signal were given and the stopwatch started. With the cursor in the center of the screen, the subject placed his or her hand on the mouse, manipulated the image in the viewport until the region in the tree where the word either was or should have been was displayed, and said either "Yes" or "No" to indicate whether the word was or was not in the tree. The experimenter then said "Good," recorded the time and whether the correct result was obtained.

After completing all the Task One training and trials, a three-minute rest break was given followed by Task Two. Again six groups of trials, one for each navi-

gational technique with and without the map window, were given. No additional training was used, since Task Two followed directly from the Task One trials.

An individual trial for Task Two proceeded as follows: the subject started with his or her hand off the mouse. A "Ready?" followed by a "Go" signal were given and timing begun. The subject manipulated the mouse to move the main display first from the apex node, to the rightmost leaf, then back--five times. The subjects were again instructed to work as quickly as possible without making errors. Only trials in which every target node was correctly located were counted. The same binary tree was used for all trials for a given technique in Task Two.

Design: The study was organized as two different three-by-two experiments, one for Task One and one for Task Two. The independent variables were the navigational technique (roam, zoom, or scroll) and the presence or absence of the map window. The dependent variable was error-free response time.

For Task One, each of the six subjects completed five trials for each combination of navigational technique with or without the map window. Thus there were a total of 60 data points for each pair-wise comparison. For Task Two, each of the six subjects completed three trials for each combination of technique with or without the map window. There were therefore a total of 36 data points for each pair-wise comparison.

Control: The order of the navigational techniques for the presence or absence of the map window was controlled to avoid learning effect. For all trials in Task One and between groups of trials in Task Two, the dictionary trees were varied to avoid having learning about individual trees affect the comparison of techniques or evaluation of the map window.

The dictionaries contained 280 words, selected at random from an on-line public-domain dictionary, but controlled for character length, number of syllables, and familiarity. About fifteen nodes of the tree were visible in the main display at one time. The target words were chosen so that about half actually were in the displayed trees. They were presented to the subjects in a random order to insure there was no known linguistic or spatial pattern in their order.

Analysis: We used analysis of variance (ANOVA) methodology to analyze the trial data. Navigational Technique and Map Window were the within-subject variables.

To test the hypothesis that navigation performance was better with a map window than without, we utilized the two-way ANOVA technique with multiple observations per cell; here it was a 3x2 table with replications. To test the hypothesis that our navigational techniques were better than scroll bars, we used 3 pairwise 2x2 ANOVA tables with replications. The first task had 30 replications per cell; the second task had 18.

Results of Task One

Six subjects were run with a total of ten trials for each technique, five with and five without the map window. All the subjects run were able to successfully complete the experiment. The results are summarized in tables below. Of the 180 timed trials, eight were discarded due to subject error.

Several of the results are significant: the use of a map window was found to be significantly better than without (90% confidence); the roam-and-zoom technique and the roam technique were found to be significantly better than scroll bars (both with 90% confidence). The experiment design did not provide for enough data points to allow pair-wise analysis of each of the individual techniques when showing the map window as compared to when it was hidden.

Table 1. Task One: Map window vs. No Map window. Map window and No Map window were the two columns; the three navigational techniques formed the three rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.10}$	F-value > $F_{0.10}$?
Column means	522.8	1	522.8	3.41	2.71	yes
Row means	641.6	2	320.8	2.09	2.30	no
Interaction	237.8	2	118.9	0.77	2.30	no
Between	1402.3	5	280.4	1.83	1.85	no
Within	26628.6	174	153.0			
Total	28030.9	179	156.5			

Table 2. Task One: Scroll Bars vs. Roam. Map window and No Map window were the two columns; the use of scroll bars and roam techniques formed the two rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.10}$	F-value > $F_{0.10}$?
Column means	108.3	1	108.3	0.66	2.71	no
Row means	498.8	1	498.8	3.07	2.71	yes
Interaction	18.7	1	18.7	0.11	2.71	no
Between	625.9	3	208.6	1.28	2.08	no
Within	18832.1	116	162.3			
Total	19458.1	119	163.5			

Table 3. Task One: Scroll Bars vs. Roam and Zoom. Map window and No Map window were the two columns; the use of scroll bars and roam/zoom techniques formed the two rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.10}$	F-value > $F_{0.10}$?
Column means	567.0	1	567.0	3.50	2.71	yes
Row means	476.1	1	476.1	2.94	2.71	yes
Interaction	154.0	1	154.0	0.95	2.71	no
Between	1197.2	3	399.0	2.46	2.08	yes
Within	18742.4	116	161.5			
Total	19939.6	119	167.5			

Table 4. Task One: Roam vs. Roam and Zoom. Map window and No Map window were the two columns; the use of roam and roam/zoom techniques formed the two rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.10}$	F-value > $F_{0.10}$?
Column means	492.6	1	492.6	3.64	2.71	yes
Row means	0.1	1	0.1	0.00	2.71	no
Interaction	180.6	1	180.6	1.33	2.71	no
Between	673.4	3	224.4	1.66	2.08	no
Within	15682.5	116	135.1			
Total	16356.0	119	137.4			

Observations from Task One

A wide variety of subject search strategies were observed. Some subjects followed a single path from the apex of the tree to the target nodes. Others went directly to the leaf nodes in the general region they expected the target word, and then systematically searched left or right and finally up until the word was located. Both of these search techniques required sequential movement, jumping movement, orthogonal movement in horizontal or vertical directions, and diagonal motion.

Having the entire tree displayed in miniature in the map window was clearly beneficial. Even though only the shape and not a node's label was visible in the map window, it still allowed a subject to move to the correct location based on what he could infer from the shape of the tree. For example, this greatly helped in locating the leaves of the tree.

With the aid of the map windows, we observed users quickly locating nodes and displaying them in the main display to see their contents. Without the map window, subjects would often take several attempts to move to a particular node and view its contents.

Because the map window was fixed, as the size of the search tree increased, the size of the individual nodes appearing in the map window and the size of

the wire-frame box denoting the main display became quite small. Therefore, for very large trees, subjects had difficulty grabbing the wire-frame in roam mode, and drawing boxes of the correct size when zooming. With the zoom technique, some subjects overcame this by zooming in the main display.

Grabbing the wire-frame box was particularly difficult when the subject only wanted to move the main display a small amount in one direction. Two roam trials were rejected because the subjects tried to use the scroll bar's page-up or page-down option, which appeared to be available on the screen. Several voiced the need for a sequential option with both the roam-and-zoom techniques. A smaller gain for the mouse might also help this problem. Several subjects mentioned that the gain was set higher than that at which they were adept.

Results of Task Two

The tables below summarizes the results of Task Two. Of the 108 trials, three were discarded due to subject error.

Several of the results are significant: the use of a map window was found to be significantly better than without (99.5% confidence); both the roam technique and the roam-and-zoom technique were found to be significantly better than scroll bars (99.5% confidence and 99.5% confidence, respectively).

Table 5. Task Two: Map window vs. No Map window. Map window and No Map window were the two columns; the three navigational techniques formed the three rows.						
	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.005}$	F-value > $F_{0.005}$?
Column means	3562.0	1	3562.0	31.96	7.88	yes
Row means	7491.8	2	3745.9	33.61	5.30	yes
Interaction	361.1	2	180.5	1.62	5.30	no
Between	11415.0	5	2283.0	20.48	3.35	yes
Within	11364.9	102	111.4			
Total	22780.0	107	212.8			

Table 6. Task Two: Scroll Bars vs. Roam. Map window and No Map window were the two columns; the use of scroll bars and roam techniques formed the two rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.005}$	F-value > $F_{0.005}$?
Column means	3270.6	1	3270.6	28.62	7.88	yes
Row means	5473.7	1	5473.7	47.90	7.88	yes
Interaction	45.9	1	45.9	0.40	7.88	no
Between	8790.3	3	2930.1	25.64	4.28	yes
Within	7769.7	68	114.2			
Total	16560.0	71	233.2			

Table 7. Task Two: Scroll Bars vs. Roam and Zoom. Map window and No Map window were the two columns; the use of scroll bars and roam/zoom techniques formed the two rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.005}$	F-value > $F_{0.005}$?
Column means	2228.9	1	2228.9	18.24	7.88	yes
Row means	5760.3	1	5760.3	47.14	7.88	yes
Interaction	423.6	1	423.6	3.46	7.88	no
Between	8412.9	3	2804.3	22.95	4.28	yes
Within	8308.0	68	122.1			
Total	16720.9	71	235.5			

Table 8. Task Two: Roam vs. Roam and Zoom. Map window and No Map window were the two columns; the use of roam and roam/zoom techniques formed the two rows.

	Sum of squares	Degrees of freedom	Mean square	F-value	$F_{0.005}$	F-value > $F_{0.005}$?
Column means	1810.5	1	1810.5	18.50	7.88	yes
Row means	3.6	1	3.6	0.03	7.88	no
Interaction	66.8	1	66.8	0.68	7.88	no
Between	1880.9	3	626.9	6.40	4.28	yes
Within	6652.1	68	97.8			
Total	8533.1	71	120.1			

Observations from Task Two

Using the page-down feature when appropriate appeared to be advantageous. Subject six discovered that the page-down feature would move the main display from showing the top of the trees to showing the bottom layers. Thus to move from apex to rightmost leaf with scroll bars required moving the cursor to the vertical scroll bar and hitting the page-down region, moving the cursor to the horizontal bar and adjusting the main display location until the rightmost leaf was in view.

Users often missed locating either the apex or rightmost leaf node on their first try. With the map window present, the primary difficulty seemed to be manually manipulating the main display rather than difficulty knowing the target location. Without the map window, additional error occurred when users tried to navigate by interpolating the target location relative to the sides of the empty map window.

Discussion

From the results of the above experiments, we can draw two conclusions: first, map windows are a significant advantage when moving in a large two-dimensional information space, particularly one that is unknown to the user; and second, the roam and the roam-and-zoom techniques each are superior to scroll bars for movement in a large two-dimensional space.

We did not have sufficient data points to determine conclusively whether roam alone was better that roam and zoom together. A simple keystroke model calculation might indicate that fewer mental cycles (Card, Moran, and Newell, 1980 and 1983) and fewer hand motions would actually be needed with the roam-and-zoom technique. But we suspected that the zoom technique required more time to be spent thinking about the level of zoom, because a bad zoom is costly (in time and mental effort) to recover from, and the choice of the exact location of the starting point is often difficult and time-consuming.

From observation, the *paging mechanism* of the scroll bars proved valuable. No users used the scrolling mechanism; however, we believe that a significantly faster scrolling mechanism on the scroll bars would have resulted in it seeing more use. Unfortunately, the hardware needed for this performance improvement is often costly.

A great deal of learning seemed to take place during the trials in Task Two where the dictionary was not changed within sets of trials for a given technique, with or without the map window. This would suggest, as one might think, that the map windows are more useful in relatively unknown, or complex information environments.

We felt the key function missing from our roam and zoom techniques was a sequential paging mechanism, similar to that found in the Microsoft *Windows* scroll bars. Such function is easily added.

An area of further work was suggested by one of the reviewers of this paper. It may well be that this study is only relevant to cases of well structured (and simply structured) graphs. But such cases are generally not the rule, at least in hypertext systems. Hypertext browsers usually contain graphs of complex, interlinked networks. Search and exploration in that setting may well be qualitatively different from navigating in sorted binary trees. For example, workers on at least one hypertext system have found map windows to be useful for gradual acquisition of familiarity with the total graph. As users repeatedly explore and search a graph, they begin to learn its "shape" as displayed in the map window. Over time, the user's cognitive model of the network connects regions of the map with contents of the nodes in those regions. Interestingly, this requires that the graph's overall shape change only locally over time. A reasonable question is how useful are maps for complex graphs that change more frequently?

Finally, we can envision other specialized two-dimensional navigational situations in which neither roam-and-zoom nor roam only are ideal. For example, some applications might require only local or only orthogonal movements, or others might involve only a relatively small information space. But given the goal of a technique for general-purpose, two-dimensional navigation, both techniques seem more than acceptable.

Acknowledgements

Trademarks

IBM® is a registered trademark and IBM PC-AT™ is a trademark of International Business Machines Corporation.

Macintosh® is a registered trademark of Apple Computer Company.

Microsoft® and MS-DOS® are registered trademarks of Microsoft Corporation.

References

Baird, J.C. Studies of the cognitive representation of spatial relations. *Journal of Experimental Psychology: General* **108** (1979) 90-106.

Beard, D.V., M.M. Mantei, and T.J. Teorey. Metaform: updatable form screens and their application to the use of office metaphors in query language instruction. *Behaviour and Information Technology* **6**, 2 (1987) 135-157.

Beard, D.V. and J. Symon. On Spatially Locating Spatial Information: A Medical Image Information System and Subject Experiment. Forthcoming (1988).

Brooks, F.P. Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings. *Proceedings of the 1986 Workshop on Interactive 3D Graphics*, Chapel Hill, NC, October 23-24, 1986, 9-21.

Brooks, F.P. No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer* **20**, 4 (April 1987) 10-19.

Bush, V. As we may think. *Atlantic Monthly* **176**, 1 (July 1945) 101-108.

Caplinger, M. Graphical Database Browsing. *SIGOIS Proceedings*, Providence, RI (October 1986) 113-119.

Card, S.K., T.P. Moran, and A. Newell. The Keystroke-Level Model for user performance time with interactive systems. *Communications of the ACM* **23** (1980) 396-410.

Card, S.K., T.P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction.* Erlbaum, Hillsdale, NJ, 1983.

Conklin, J. Hypertext: An Introduction and Survey. *IEEE Computer* **20**, 9 (September 1987) 17-41.

Delisle, N. and M. Schwartz. Neptune: A Hypertext System for CAD Applications. *Proceedings of ACM SIGMOD International Conference on Management of Data*, Washington, DC, May 28-30, 1986, 132-143.

Donelson, W. Spatial Management of Information. *Computer Graphics* **12**, 3 (August 1978) 203-209. (Proceedings of SIGGRAPH '78, Atlanta, GA)

Douglas, S.A. and T.P. Moran. Learning text editor semantics by analogy. *Proceedings of the CHI '83 Conference on Human Factors in Computing Systems.*

Dumais, S.T. and W.P. Jones. A comparison of symbolic and spatial filing. *Proceedings of the CHI '85 Conference on Human Factors in Computing Systems*, (April 1985) 127-130.

Englebart, D.C. A conceptual framework for the augmentation of man's intellect. In *Vistas in Information Handling, Vol. 1*, P.D. Howerton and D.C. Weeks, eds. Spartan Books, Washington, DC, 1963.

Englebart, D.C. and W.K. English. A research center for augmenting human intellect. *AFIPS Proceedings, Fall Joint Computer Conference* **33** (1968) 395-410.

Fairbairn, D.G. and J.H. Rowson. ICARUS: An interactive integrated circuit layout program. *Proceedings of the 15th Annual Design Automation Conference, IEEE*, (1978) 188-192.

Fitts, P.M. The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology* **47** (1954) 381-391.

Foss, D., M. Rosson, and P. Smith. Reducing Manual Labor: An experimental analysis of learning aids for a text editor. *Proceedings of the Gaithersburg conference on human factors in computing systems*. Washington, DC, National Bureau of Standards, 1982, 332-336.

Frank, A. MAPQUERY: Data Base Query Language for Retrieval of Geometric Data and their Graphical Representation. *Computer Graphics* **16**, 3 (July 1982) 199-207.

Furnas, G.W. Generalized Fisheye Views. *Proceedings of the CHI '86 Conference on Human Factors in Computing Systems*, (April 1986) 16-23.

Griffith, R.L. Three Principles of Representation for Semantic Networks. *ACM Transactions on Database Systems* **7**, 3 (September 1982) 417-442.

Halasz, F.G. and T.P. Moran. Analogy considered harmful. *Proceedings of the Gaithersburg conference on human factors in computing systems*. Washington, DC, National Bureau of Standards, 1982.

Halasz, F.G., T.P. Moran, and R.H. Trigg. NoteCards in a Nutshell. *CHI '87 and GI Proceedings*. (April 1987) 45-52.

Hayes, B. Rank-and-file thinking. *Lotus*, (June 1985) 73-77.

Herot, C.F. Spatial Management of Data. *ACM Transactions on Database Systems* **5**, 4 (December 1980) 493-513.

Hypertext '87 Papers, J.B. Smith and F.G. Halasz, eds., Chapel Hill, NC, November 13-15, 1987.

Kay, A. Microelectronics and the personal computer. *Scientific American*, (September 1977) 387-403.

Mantei, M.M. *Disorientation Behavior in Human-Computer Interaction*. Ph.D. Dissertation, University of Southern California, Los Angeles, CA, 1982.

Microsoft Corporation. *Microsoft Windows Software Development Kit, Programmer's Reference.* Document Number 050051053-100-100-1185, Bellevue, WA, 1985.

Nelson, T.H. The Hypertext. *Proceedings of the International Documentation Federation Annual Conference,* 1965.

Quillian, R. Semantic memory. In *Semantic Information Processing,* M. Minsky, ed. MIT Press, Cambridge, MA, 1968, 227-268.

Rumelhart, D. and D. Norman. Analogical processes in learning. In *Cognitive Skills and Their Acquisition,* J. Anderson, ed. Hillsdale, NJ, Erlbaum, 1981, 335-359.

Schneiderman, B. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer* **16,** 8 (August 1983) 57-69.

Schneiderman, B. *Designing the User Interface.* Addison-Wesley, Reading, MA, 1987.

Smith, J.B. and C.F. Smith. *Conceptual Tools for Writing.* Technical Report TR87-024, Department of Computer Science, University of North Carolina at Chapel Hill, October 1987.

Smith, J.B., S.F. Weiss, M. Lansman, J.D. Bolter, and D.V. Beard. *An Experimental Study of Writers' Cognitive Strategies Using Advanced Computer Tools.* Project description submitted to the U.S. Army Research Institute for the Behavioral and Social Sciences. Department of Computer Science, University of North Carolina at Chapel Hill, September 1985.

Smith, J.B., S.F. Weiss, G.F. Ferguson, M. Lansman, J.D. Bolter, and D.V. Beard. *WE: A Writing Environment for Professionals.* Technical Report TR86-025, Department of Computer Science, University of North Carolina at Chapel Hill, August 1986.

Welford, A.T. *Fundamentals of Skill.* Menthuen, London, 1968.

Young, R. The machine inside the machine: User's models of pocket calculators. *International Journal of Man-Machine Studies* **15** (1981) 51-85.