Window Control Strategies for On-Line Text Traversal

Kobi Lifshitz and Ben Shneiderman*
Department of Computer Science
and
Human-Computer Interaction Laboratory
University of Maryland, College Park, MD 20742

July 30, 1987

Abstract

Larger and higher resolution screens that support multiple windows are now widely available. They are often used to traverse hypertext databases for fact retrieval, education or casual browsing. This paper introduces window control strategies that we implemented, describes their cognitive complexity, and characterizes the tasks that users might encounter. An informal usage by dozens of visitors and a usability test with four subjects performing information search tasks revealed the limitations of several strategies and guided us to select user control of article placement in a tiled non-overlapping multiple window display.

*Address correspondence to Ben Shneiderman
PART I: Introduction

In the past, textual information was displayed on paper and was organized by pages, articles, chapters or books. When text began to be displayed on video display terminals, the VDT's small size and poor resolution slowed reading by as much as 30% when compared to paper [Muter et al., 1982]. A screen page accommodated less than half the text that a printed page could, and as a result required the user to change his cognitive task from reading the text to manipulating it very frequently. With the introduction of large and high resolution screens, where the display may handle more text with easily readable black letters on a white background, the interest in computer browsing is growing. We explored multi-window browsing strategies that might prove advantageous for on-line text traversal. We first encountered the issue while converting Hyperties (a hypertext system based on The Interactive Encyclopedia System that used small personal computer (PC) displays) to run on the SUN 3/50 workstation.

1. Hyperties - The Interactive Encyclopedia System

Hyperties is a hypertext system that was originally developed to serve novice users in a museum environment. It displays articles in which keywords are highlighted by a special attribute (usually a color when it is available on the PC or a bold font for monochrome or the SUN) thus creating an embedded menu. When users choose a keyword with any appropriate input device (mouse, jump arrow keys, or touch) a definition about the selected term is displayed at the bottom of the screen and a new command FULL ARTICLE appears. Users can select the FULL ARTICLE command to access a new article about the selected keyword, which replaces the previous article. Within articles the user can flip through the pages or see definitions of highlighted terms. Among articles, users can select new articles, follow articles back in the path they had just created, or view the index for the whole encyclopedia from which they also may select keywords [Hyperties User's Manual].

The PC version (Figure 1) was developed in a non-window environment. With the default setting of double spacing, only nine lines of text appear per page with 68 characters per line and 6 character margins. The rest of the lines are dedicated to additional features. The article name is displayed on the top left and the page numbering on the top right. Below the article one dashed line separates the text from the definition area. Three lines are reserved for the definitions and the bottom two lines are used for commands like NEXT PAGE, BACK PAGE, FULL ARTICLE, RETURN TO, INDEX and QUIT.

2. Related work

2.a Window management

It was suggested [Bury et al., 1985] that the benefits of multiple windows reduce the load of the user's short-term memory, resulting in freeing cognitive resources to the tasks the user is trying to accomplish. In a study comparing a window system to non-window system, Bury et al., produced some conflicting, nonetheless interesting results. While overall performance was poorer for the window system, an analysis of the component breakdown showed that time spent in the task domain was shorter for the window system. The rest of the time was largely spent on screen arrangements.
PLACES: AUSTRIA

Austria (see map) holds a special place in the history of the Holocaust.

Situated between Eastern and Western Europe, possessing a vibrant and culturally creative Jewish community on the eve of World War II, Austria had also provided the young Adolf Hitler, himself an Austrian raised near Linz, with important lessons in the political uses of antisemitism. Leading Nazis came from Austria: the names of Adolf Hitler, Adolf Eichmann, who organized the deportations of the Jews to the death camps, and Ernst Kaltenbrunner, the head of the Reich Main Office for Security, 1943-45, readily come to mind. As

Linz - city in northern Austria; childhood home of Adolf Hitler and other leading Nazis

The two widely used window management schemes are tiled and overlapping. Until recently, many believed that overlapping windows were better than tiled windows because they allowed users to flexibly control screen arrangement.

Bly and Rosenberg [Bly et al., 1985] studied subjects performing two tasks, regular and irregular, in each of the two windowing schemes. Both types of tasks had to do with matching graphical objects with their textual description. In the regular task both objects and text were arranged in an orderly manner at the top of their corresponding documents. In the irregular task both objects and text were distributed throughout the document randomly. Bly and Rosenberg believed that the regular task would relieve users from window management, therefore favoring the tiled windows. As for the irregular task they believed users would try to conform windows to the contents, and therefore would favor the overlapping windows. After eliminating some of the results and considering only inexperienced users they found that tiled windows beat overlapping windows in the speed of task completion and reduced the number of window manipulations that users performed during the experiment. Even though experienced users of overlapping windows performed faster when working on the irregular task with overlapping windows, they believed that "for a given level of expertise, tiled windows are better in both situations." These results support the previous observation by Bury et al. that overlapping windows require increased window manipulations.

Tiled window management can by itself be broken into two categories: fixed tiling and adjustable tiling. Fixed tiling refers to systems where the windows are
given an initial position which is unchanged throughout the interaction. Adjustable
tiling, on the other hand, allows the change in position and shape of the initial
window set up under the constraint that all windows are fully visible and no
overlapping occurs. For a discussion of a variety of "cognitive layouts" in multi
window systems see Norman, Weldon, and Shneiderman (1986).

2.b Hypertext systems

A number of project teams have created hypertext systems which differ in their
application, simplicity of user interface and authorship mechanism.

Notecards [Trigg et al., 1986] [Halasz et al., 1987] was developed at the
Intelligent Systems Lab of Xerox PARC and addresses "system supported
collaboration at the social interaction level." Previous systems were usually
designed as single-user systems without interaction among authors and readers.
Once the authors had completed their writing, readers could browse through the
database without the ability to add their own annotation. In contrast, Notecards
does not distinguish between authors and readers. All users have the same access
rights and are able to add or modify existing material in the database.

Notecards operates in a multiple overlapping window environment where each
card is represented by a separate window. This added feature while allowing for
flexible interaction, is mainly intended for knowledgeable users who are experienced
with systems where they had to manipulate windows. Novice users will have a hard
time operating the system because of the added complexity and the amount of user
control assumed for these operations.

Intermedia [Garrett et al., 1986] [Yankelovich et al., 1985] was developed at
Brown University's Institute for Research in Information and Scholarship. As in
Notecards there is no distinction between authors and readers. Although one might
only want to browse through the system, all interfaces were designed the same way,
and do not take into account novice users. Intermedia is implemented in a multiple
overlapping window environment where several applications can run
simultaneously. References may not be deterministic, as some words may be part of
several links. In these cases, additional dialogue is required between the user and
the system prior to any link traversal. This feature while adding to the flexibility,
diverts the user's attention from the task domain.

GUIDE, [O'Bannon 1987] a hypertext system on the Macintosh, is an
authoring/browsing system. MINIGUIDE, which is a subset of GUIDE is available
for browsing only. Readers can alter the appearance of buttons and links, or
eliminate them altogether.

PART II: Browsing Strategies

While some users will be pleased to have the full features of multiple
overlapping windows, this strategy requires substantial cognitive overhead of users'
management of the windows (size, placement, replacement of contents, elimination
of window, etc.). By contrast a single window approach is the utmost in simplicity,
and although users have been satisfied with it, the attraction of multiple windows is
great. Designers and user are not limited in choosing from these extremes; there are
many intermediate possibilities worthy of exploration.
1. Single Window

The single window strategy where users can only see one page of a single article at a time (Figure 2.) preserves the original style of the Hyperties PC version.

![Figure 2. A single window environment. The top subwindow displays the article, the middle subwindow displays the definition and the bottom subwindow displays permissible operations.]

However, on the SUN we could enlarge the visual scope by displaying 34 lines with 60 characters per line and 6 character margins. While we did not allow users to manipulate the window, this strategy serves the purpose of browsing. Taking into account our well learned ability to read linear text, it seems that the single window strategy is sufficient for exploratory browsing or for slightly more challenging tasks requiring only forward referencing.

The limitation of such a strategy arises when more complex tasks are encountered. In a medical encyclopedia containing prescription drug information, for example, a physician might want to look at several alternatives and compare them or their interactions with previously prescribed medications. Certainly a single window would be limiting. The inability to quickly jump from one reference to another and the enormous overhead that will be required by the user to switch among the desired articles will make the system tedious and ineffective. The nature of that problem and many others leads us to think about numerous approaches which include a number (N) of windows.

Before we proceed let us establish a graphical notation that will be used throughout this paper to describe the different states of browsing strategies. Solid links and nodes represent a path the user had already followed, while dotted links and nodes represent valid commands the user can select. Circles stand for pages of articles and are labeled with an article and page identifier (e.g. B3 represents article article B page 3). The bold rectangle represents the user's current view, i.e. the pages that are visible on the screen in the current state. Unless mentioned otherwise, definitions and the index page are not described in the graph and are always available as a command from the active window. An active window is a window from which all the operations that were originally developed for the single window version are permissible. In the graphical presentation, nodes that are displayed in the active window are shaded in gray.

Figure 3 represents two possible situations of a single window environment. In this example (a) represents the state where the first page of a second article is
Figure 3. Path traversal of a single window screen. (BP = back page, NP = next page, FA = new full article, RT = return to previous article)

currently present on the screen and is fully active. In (b) the third page of the first article is visible and fully active.

2. Multiple windows (N = k, is constant)

First we examine strategies for two fixed non-overlapped windows, before proceeding to multiple windows.

2.a Two Windows (N = 2)

Since we were working with the SUN workstation, whose screen allows the display of almost two double-spaced pages, the most natural extension included two side by side windows (Figure 4). We limited the user's ability to manipulate and resize windows in order to reduce confusion and distraction in reading its contents, thus maximizing the amount of text displayed within the two windows. Instead we were more concerned with different browsing control strategies, and the circumstances under which one strategy may be superior to the other. In the following discussion on panning, the active window appears on the right window, while the background window appears on the left window (Figure 5).

2.a.1 Panning-by-pages

The concept of panning-by-pages identifies the main entity of the system as a page of an article. When the user selects any operation that refreshes the active (right) window the previous contents of the right window moves to the left and a new page is displayed on the right. The new page can be the next page of the current article, the first page of a new article or it can be the index. (Index pages do not move across windows the same way as article pages because the index is always accessible from the active window). Once a page is displayed on the background left window, no commands are allowed to be performed from there, such as NEXT PAGE or FULL
The idea of Anschluss or union between Austria and Germany was deeply rooted among Germans on both sides of the Austrian-German border long before the rise of Nazism. It is an idea as old as German nationalism itself, an idea inseparable from the German nationalist dream of creating a pan-German or Greater German state. Austrian support for Anschluss was particularly strong during the months immediately following World War I, a time when Austrians feared that an Austria stripped of her empire could not survive. The victorious Allies disapproved of such a union and specifically forbade it in both the Treaty of Versailles and the Treaty of St. Germain.

Austrian nationalism remained weak throughout the interwar period (1918-38). During those years Austria, like Germany, gave rise to a number of right-wing and fascist political movements. Indeed, Adolf Hitler's Nazi Party had a sizable Austrian branch. In 1934, Engelbert Dollfuss, a member of the Christian Social Party, destroyed the First Republic's fragile parliamentary democracy and established a right-wing dictatorship. Throughout the life of this right-wing government, Austrian Nazis prepared the way for Anschluss by capitalizing on latent German nationalism.

Support for Anschluss, of course, was not universal. Even among right-wing elements, many Austrians demonstrated their reluctance to submerge Austria into a Nazi-led state. As late as March 9, 1938, Chancellor Kurt von Schuschnigg attempted to forestall Anschluss by calling for a national plebiscite on the question of Austrian independence.

Two days later, on March 11, Adolf Hitler responded to Schuschnigg's action by issuing an ultimatum that the plebiscite be postponed. On March 12, Hitler dispatched troops into Austria on the grounds that Schuschnigg's government could no longer maintain order. Schuschnigg resigned his post on March 11 and Arthur Seyss-Inquart, an Austrian Nazi, quickly assumed the chancellorship.

Figure 4. A sample screen of Hyperties on the SUN 3/50 workstation

Figure 5. Window layout that was used on the SUN for the pilot study

ARTICLE. We also did not allow the display of definitions on the background window, although someone else might have chosen to do otherwise.

This strategy allows users to see more text than in the single window environment. In the case where two consecutive pages of the same article are displayed this strategy allows for a continuous flow of reading which is not available with the single window control strategy. In the case where a new article is selected,
the background page serves as a reminder of the context from which the new article was invoked.

The problem with this strategy is that a page has no semantic meaning. Pages are the result of system dependent screen constraints and have nothing to do with meaningful logical structures. It seems more natural to view a document in a manner that closely resembles the logical structure of the system. Screen inertia is another important issue. Users might be overwhelmed by the frequent movement of the text around the windows which results from any permissible action, in particular when users trace their way back through the path they had just taken. On one hand, if the designer allows them to return to the previous article from any page, then the regression from the path would skip through a number of pages, creating a discontinuity in the backward panning. On the other hand if continuity is desired then the designer puts the burden on the user to trace back through every page visited, making back jumps a lengthy process.

Figure 6 represents two possible situations of a double window environment.

(a) 
(b)

Figure 6. Path traversal of a two window screen using panning-by-pages. (BP = back page, NP = next page, FA = new full article, RT = return to previous article)

Example (a) represents the case where the background window (unshaded A2) displays the second page of the first article and the active window (shaded B1) displays the first page of a second article. In (b) the background window displays the same second page as in (a) but the active window displays the third page of the same article.

2.a.2 Panning-by-articles, background is inactive

When panning-by-articles each window presents a single article of the hypertext system. From the path implementation point of view the active window represents the last visited node, while the background represents the previously visited node. In this version the background window is used as a snapshot of the last state of the previously visited article, just before the new selection was made, and like in the panning-by-pages case it is completely inactive.
In addition to the advantage of having more text appear on the screen, as in panning-by-pages, this strategy enforces context connectivity between the two nodes. This browsing technique facilitates some comparative tasks. Using our previous example, the physician might read an article of one drug first and as the article makes a reference to another drug that may be used as a substitute to the original drug, the physician may access that new article. Since the old article still appears in the background, the physician is able to compare them both before making a decision.

Screen inertia is also considerably improved since the more frequently used NEXT and BACK PAGE commands change only one window and a complete screen rearrangement occurs only on the FULL ARTICLE and RETURN TO commands.

A major drawback to this strategy is the inability to page through the background article. In particular, when the reference appears at the very top or bottom of the page, there might be cases where the paragraph from which the reference was made is not fully visible. In these cases the advantage of connectivity might be lost as the reference is taken out of its context. Another disadvantage of this method is the inability to access definitions from the left window, although the need for some short descriptions may enhance the understanding of the reading material.

Figure 7 represents two possible situations of the panning-by-articles strategy.

![Diagram of panning-by-articles strategy](image)

**Figure 7.** Path traversal of a two window screen using panning by articles with an inactive background. (BP = back page, NP = next page, FA = new full article, RT = return to previous article)

7(a) is no different from the example in Figure 6(a). Note however the progressive change in 7(b) after paging through the second article. While the active window points to the second page of the second article, the background window still points to the same page of the previous article (as it did in 7(a) when the first page of the second article was placed in the active window) and the first page of the current article no longer visible (but still accessible, as the BP link indicates).
2.a.3 Panning-by-articles, background is partially active

To improve on the previous strategy, we relax most of the restrictions that were imposed on the inactive background window. All commands are accepted on the background window except for the full article. While these improvements eliminate most of the deficiencies of the previous strategy, the user might lose the connectivity between articles, especially if the background article is very long and the user has turned too many pages.

This browsing strategy very much resembles tree traversal and seems to favor tasks that require a tree-like cognitive approach for problem solving. For example, a task may require the user to identify a group, such as the authors of the constitution, and then for each to find the state they represented; the desired procedure would be to locate the group within the database and then while this article is mainly in the background the user will jump back and forth among the nodes that describe each of the members and the node that describes the group. Figure 8 represents two possible situations of the partially inactive strategy. Both examples are the extensions of the examples in figure 7. Note that the only changes are the addition of possible paging from the background window.

2.a.4 Panning in zigzags, only one window is active

The three panning strategies we described earlier use a fixed location for the active window. In the zigzag strategy new pages or articles (depending on whether we talk about zigzag-by-pages or zigzag-by-articles) are placed over the least recently refreshed window. Instead of being in a fixed position, the current active window becomes passive and the new entity (page or article) is placed in the previous background window which now becomes active. One advantage to this strategy is that text which appears on the screen does not have to be moved around, supporting the notion of screen inertia and eliminating some confusion. The disadvantage of
this strategy is the need for the user to remember which window is the active one. This drawback can be ameliorated by special marking of the active window, with a wider border or a different color, for example.

2.a.5 Independent windows

The independent windows strategy allows for two separate single window systems to run side by side. Both are active and all functions within each window are accessible. Replacement articles stay within the window where the keyword was initially referenced. This strategy seems to benefit parallel browsing of articles that might have some information in common, comparisons of two different subjects, or even finding complementary information that might not be within the scope of the article initially visited.

The advantage of independent browsing is that when users finally find a needed article, they do not have to replace it while searching for another related article. Instead, the user can shift attention to the other window and without affecting the status of the first window find the additional information. One problem associated with this strategy is that moving the mouse across the large window is sometimes annoying. Also, when an article has already been accessed in one window has a reference to the next desired article, instead of utilizing the second window, the user might be tempted to use the currently visible reference and therefore lose the referencing article.

This might happen because reading (as described in the single window issues) is sequential and parallelism might disturb the users. Figure 9 describes a possible situation of two independent windows. Note the similarity to the example in figure 2. Here too, each of the independent windows represents a valid path of the single window system. The only difference, however, is that now the user can view them in parallel.

![Diagram](attachment:diagram.png)

Figure 9. Path traversal of a two window screen using independent browsing. (BP = back page, NP = next page, FA = new full article, RT = return to previous article)
2.a.6 Permanent Index Page

In this strategy, rather than having the index only as a selection command, the background window is completely devoted to the index. Highlighted terms can be selected from both windows. For both windows definitions appear on the same side as the selected term, however, when a full article is selected from either window, the new article is always placed in the active window, replacing the previous article which disappears from the screen.

This strategy reduces in half the amount of actual text that appears on the screen. While some might see that as a weakness, the ability to have the index visible at all times enables the user to have a clear overview of the database and available topics.

A study of a hierarchical browser for PASCAL programs [Shneiderman et al., 1986] showed that a hierarchical view of the programs reduced the number string searches, reduced the time to answer comprehension questions and increased users' satisfaction. Their results suggest that such a high level view will benefit users of other applications such as databases, knowledge bases, text, and visual informations such as maps.

In our environment, when someone wants to browse for some topic but is not sure as to the way the topic is referenced, the index may provide a constant reminder for possible choices. The constant presence of the index, with the additional information of how many articles exist in the database and the topics available, keeps users aware at all times of their moves and reduces the possibility of disorientation.

2.a.7 Visible Definition Page

In this strategy the background window is devoted to the display of definitions for all the keywords that appear on the current page. The idea, which is borrowed from some science textbooks, enables users to see the short explanations on the screen with only a quick eye movement. In the absence of the definitions, users are required to shift attention to an external input device and make a few actions before being able to access the definition. Extra actions break the user's concentration and might lead to unnecessary frustration and mistakes. The constant presence of the definitions also eliminates the need for the two step process now required for the access of a new article, making the system faster and easier to use.

2.a.8 User Controlled Placement

All the strategies described above concentrated on system control. While the user selects the command to be performed, the resulting placements of subsequent articles is forced upon the user. The following strategy transfers control to users, who can place the new article at the window of their choice. During the selection of a full article on a certain topic, users will press the right or left buttons to make the selection and identify the destination of the next article. When the user presses the right button, the article will be displayed in the right window and conversely when the user presses the left button the article will be displayed in the left window. Definitions and paging will still be displayed in the old manner, i.e. on the same side as the reference that was selected.

Putting the users in control, while maybe requiring a little more training, facilitates a number of improvements. First the users arrange the articles in a style
that is familiar to them. Second, the movement of articles on the screen is directed by the users and therefore they are not disrupted by unpredicted movements.

The greatest benefit of this method is its flexibility. Users can invoke commands so as to mimic a number of previously described strategies. Independent browsing is achieved by selecting full article references from the right window with the right button and selecting full article references from the left window with the left button. Fixed index page is achieved by placing the index on one side and all references from the index are selected with the opposite button to the index window (right button for left window index and vice versa). A zigzag pattern can be achieved by selecting everything on the right with the left button and everything on the left with the right button.

The high level of flexibility is very desirable, however, it leads to new concerns. Returning to previously seen article is ambiguous. Should each window have its own stack representing its path or should a crisscross return also be allowed? It seems to us that a different approach should be taken. Rather than thinking only in narrow terms of returning to previous article we now look at states of the hypertext, where the meaning of RETURN TO is a return to the previous state. Under that interpretation the user has less problem of understanding the meaning of the action, since the previous visible state is unique.

2.b Multiple Predetermined windows (k = constant)

The choice of two windows was influenced by the fact that each window's size resembled a double-spaced printed page. Any number of fixed windows may be acceptable provided that the text is readable from these windows, and there is a good underlying reason for the benefits of choosing one strategy over the other. Possible options for multiple windows may have four, six and eight windows lined up equally on both sides of the screen or nine windows for a square format (Figure 10.). Besides

![Diagram of fixed multiple windows](image)

Figure 10. Fixed multiple windows.

the advantages and deficiencies that we introduced for different two-window strategies, window ordering is an additional problem we have to face when we talk about multiple windows. In the discussion of multiple windows we try to generalize some of the basic strategies and issues that we have mentioned earlier.
2.b.1 Panning

Regardless of the manipulative ability of the background window, filling order can be done in several ways. One choice is to fix the location of the active window. Whenever a new article is selected it always updates the active window. Articles in background windows move in a way that an older article gives way to newer article by one access cycle and the oldest article disappears from the screen. The simplicity of identifying the most and least recent articles gives a sense of ordering to the user. There are several possibilities of ordering the windows on the screen: right to left and top to bottom where columns take precedence, rows take precedence, or even a circular way from top right to bottom right and then from bottom left to top left. Other options are limited only by the reader's mind.

2.b.2 Rotations

The rotation is the modification of the zigzag strategy used with the two window screen to any number of windows. In this strategy the active window changes its position constantly. When an article is selected it replaces the article that had been displayed for the longest time. Such a strategy eliminates the enormous amount of text movement which is more substantial with a large number of windows. The problem of locating the new active window can be solved by a thicker border, or by having an ordering field on the window header. Since the rotation can be implemented in several ways such as circular clockwise vs. circular counterclockwise, left to right vs. right to left and top to bottom vs. bottom to top, it is advisable to devise additional ways to distinguish the active window from the rest of the windows.

2.b.3 Fixed Functional Windows

The same way we presented fixed index and definition windows in the two window environment, we can add these windows to a multi-window environment, where there is more room for variations. An index window can be present, a definition window can be present or even both. It is advisable to assign these windows to dedicated areas of the screen because of their different nature which is more static than the article windows.

2.b.4 User Controlled Placement

The power of a mixed strategy that puts the user in control can be applied in a multiple window environment with slight modifications. When a highlighted term is selected, a small icon would appear. There is a limited number of ways to choose the destination of the newly selected article. One approach can subdivide the icon into areas that represents the screen window layout. Whenever one of these areas is selected the full article is placed in the corresponding window. Another method can use the object-action method used in the Macintosh or the Xerox STAR computers. The highlighted term will first be selected (maybe a change in color or icon background will acknowledge that selection) and then an icon will be moved around to the destination window. Novice users have problems moving pointing devices, particularly the mouse, and the smaller the region that is available for selection, the harder it is for them to hit the target correctly. In addition, the division of the icon into subregions makes all selectable regions close to one another making it more likely to hit the wrong targets.

The benefits of user control have been discussed earlier, but now the window management issues are more noticeable. Unless the users keep track of the articles
they place, the flexibility can introduce new difficulties. The more windows the system allows, the harder it is for the user to manage them.

3 Variable Size Windows

So far all the strategies we have described have used a prearranged window set. The final and most flexible strategy puts complete control in the user’s hands. For each new article the user selects, a new window is opened. The window can be moved, stretched, and placed at a convenient location. While this strategy supports the desktop metaphor there are a lot of issues to be resolved. First there is the issue of the initial form and placement of the new window. Should it start as an icon waiting to be opened or should it be placed open. If so, should it be placed at random or should it be in some relation to the other windows? How large should the new window be? Should it have a standard default size or should it be a function of the length of the article? Other issues may include fonts, number of lines and width of the window.

PART III: Informal Usability Test

1. Experience

Dozens of visitors to our lab as well as graduate students and faculty tried various of the window control strategies. Their informal exploration led to numerous, sometime conflicting comments and many thoughtful suggestions.

An informal usability test was conducted in order to filter the numerous strategies we have described so far. The four subjects who volunteered were encouraged to make comments during the experiment and report the mental process they had gone through.

Three distinct strategies were chosen: single window, panning-by-article background is partially active, and independent windows. In each environment the subjects performed three information search tasks against a database on "The Holocaust in Austria" written by Marsha Rozenblit and Darla Courtney. The database contains 106 articles of 50 - 2000 words.

A sequential task that involved a set of simple questions such as "How many nations attended the Evian Conference?" or "Where was the first concentration camp?" The sequence was constructed in a way that required only forward references. By this we mean that a keyword for the solution of the next problem existed in close proximity to the solution of the current problem.

A tree-like task involved more tightly connected problems such as "Name five members of the Austrian resistance movement. Also find to what political party each of them belonged." In this type of task subjects had to identify a group with a common attribute, and then branch out and find specific information for each member in the group.

The third type of task involved comparisons of two individuals or organizations such as "Compare and contrast the Central Office For Jewish Immigration and the Palestine Office." In this type of task subjects had to divide their attention equally between the two articles that described the main objects.

We had the following conjectures: the simple sequential task would be performed equally well in all browsing conditions due to the simplicity of the task
and the need to see only one article at a time. For the tree-like problems we believed that the panning-by-articles with background partially active would prove beneficial because of the connectivity that exists between the group and its members. Finally, we believed that the comparison tasks would favor the independent windows because of the need to have both articles fully accessible for browsing simultaneously.

Subjects worked with each window control strategy for 20 - 30 minutes. From the reaction of the subjects, however, it seems that they attached more importance to the number of windows present on the screen and to the window layout, rather than for a particular browsing strategy. They particularly were overwhelmed by the tight system control and were sometimes unaware where the new text was placed. Finally, all of the participants expressed their wish to be able to control the destination placement of newly displayed articles.

2. Planning

The direction for the next experiment is to redesign the article placement control. Users will be given the opportunity to point to the destination window on a small map of the system's window layout with the mouse. This will enable readers to arrange the articles according to their preference and task.

Conclusion

This paper focuses on window control strategies in textual browsing. Finding a balance between power and convenience and between user control and automaticity is an important goal for hypertext system designers. The traditional single window approach can be improved on, but the user managed cluttered desk seem an unnecessary distraction for most users in most situations. Middle of the road strategies that provide automatic or semi-automatic control of multiple windows are possible.
Bibliography

[Bly et al 1986]

[Brown 1986]
Brown, P. J. "Viewing Documents on a Screen." Kent University, 1986.

[Bury et. al. 1985]

[Cohen 1987]

[Conklin 1986]

[Delisle & Schwartz 1986]

[Garret at. al. 1986]

[Gaylin 1986]

[Haas & Hansen 1985]

[Muter et. al. 1982]

[Norman et. al. 1985]

[Norman et. al. 1986]

[O'Brien 1985]
[Shneiderman et. al. 1986]

[Trigg et. al. 1986]

[Yankelovich et. al. 1985]