A Comparison of Tiled and Overlapping Windows

Sara A. Bly Jarrett K. Rosenberg

Xerox Office Systems Division
2100 Geng Road
Palo Alto, CA 94303

Abstract

It is widely believed that overlapping windows are preferable to tiled (non-overlapping) ones, but there is very little research to support that belief. An analysis of the basic characteristics of windowing regimes predicts that there are, in fact, situations where overlapping windows are inferior to tiled. An experiment to test this prediction verified that there are indeed tasks and users for which tiled windows yield faster performance. This result suggests a need for closer study of the principles underlying windowing regimes, so that designers have a better understanding of the tradeoffs involved in using them.

1. Introduction

The current trend toward the use of overlapping windows in multi-window systems is based on the assumption that overlapping windows are clearly more beneficial to users than tiled ones. However, there is very little analysis or experimentation on window systems (though see Card *et al.*, 1984; Cohen *et al.*, 1985; Bury *et al.*, 1985). This lack of information makes it difficult to understand the relative value of different window mangagement systems. In this paper we present a simple analysis of windowing regimes and the results of an experiment to test predictions derived from the analysis.

2. An analyis of windowing regimes.

The fundamental characteristics of a multi-window system are 1) its mechanisms for window location and size, 2) the effects of manipulating one window on other windows, and 3) the degree of user or system control of these parameters. We define a *tiled* window system as one in which any open window is always fully visible; windows are not allowed to overlap. The system attempts to manage the window locations, sizes, and side-effects to maximize the use of the screen space while keeping window contents visible. Such a system typically determines the location and size of each window. When the location and/or size of a window changes, other windows are relocated and resized as needed, but never obscured. Since window size must decrease with an increase in the number of windows, a user often does not see much of the contents of a newly opened window. The decreasing size also means that a tiled system limits the number of windows that may be open simultaneously. Examples of tiled window systems include the Star system (Smith *et al.*, 1982a,b), the Cedar system (Teitelman, 1984), the Andrew system (ITC, 1984), and MicroSoft windows (Lemmons, 1983b).

We define an *overlapping* window system as one in which the user manages a window's location and size in any way desired. Thus, the user controls the use of the screen space and the visibility of window contents. When the location and/or size of a window changes, other windows may be obscured, but their locations and sizes do not change. Because windows may overlap, a user can always choose to see a full screen's worth of contents of a newly opened window. Examples of overlapping window systems include various programming environments, e.g., Smalltalk (Tesler, 1981) and Interlisp-D (Sheil, 1983), as well as the VisiOn system (Lemmons, 1983a) and WHIM (Goodfellow, 1985).

User requirements for a multi-window system can be characterized as

- the ability of windows to conform to their contents so as to maximize the the visibility of those contents, and
- the ability of the system to relieve the user of having to manage the size and location of the windows.

Overlapping window systems maximize the first user requirement. In an overlapping system, the user has control over moving, resizing, and overlapping windows anywhere on the screen. Such systems typically do very little to aid the user in window management. Tiled window systems maximize the second user requirement. Because most tiled systems attempt to satisfy the conformance requirement by using all of the available screen space, windows often change size and location when other windows are opened or closed. Tiled systems typically do very little to conform to window contents that are not of a default shape, and as the number of open windows increases, the size of each window decreases.[1]

3. Some predictions.

From the analysis above, we can predict situations in which one or the other window system is preferable. A tiled system aids a user when a) the contents of the windows basically conform to the defined arrangement of window size and placement, or b) the user does not wish to be distracted by managing the windows. An overlapping system aids a user when a) the contents of the windows do not conform to any pre-determined window arrangement, or b) the user wishes to control the window environment completely.

A task in which we expect a tiled window system to aid user performance would require that the contents of the windows have maximum visibility within the default window location and size (see Figure 1). A user would have no need for additional window management and could proceed directly with the task. Performing the same task in an overlapping system would require that the user manipulate the windows into a state similar to that automatically provided in a tiled system.

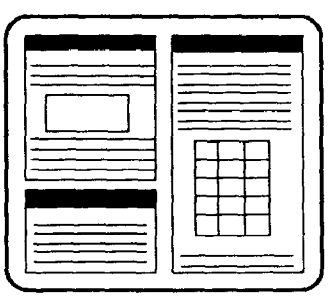

Figure 1. A task suitable for tiled windows.

Conversely, a task in which we would expect an overlapping system to aid user performance would involve windows whose contents did not conform to a regular pattern (see Figure 2). In order to obtain maximum visibility of the contents, the user would have to arrange the windows in some irregular manner, utilizing the ability to manage and overlap the windows.

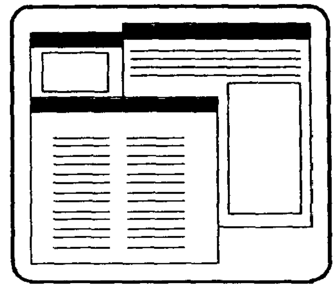

Figure 2. A task suitable for overlapping.

However, even with a task which required window manipulation in order to be performed efficiently, overlapping windows might not aid users unskilled in the use of the window management mechanisms. Such inexpert users might spend more effort in trying to master the functionality of overlapping windows than in scrolling and opening closing the windows of a tiled sytem.

4. An experiment.

In order to test the prediction that different window systems may be preferable for different tasks, we conducted an experiment using the Xerox ViewPoint Office System (a direct descendant of the Star system). The ViewPoint system was useful because the user may specify whether window management is tiled or overlapping without affecting other aspects of the system.

The ViewPoint tiled window system allows at most six open windows, three vertically on each half of the screen. When a user opens a window, it will open in the next available space. As a window opens, it divides the vertical space equally with the other windows on the same side of the screen (see Figure 3). The user may control the height of a window and whether the window is on the left or right half of the screen. In general, when a window is closed, remaining open windows attempt to fill the vacated space. If the user has manipulated windows by changing their lengths or positions in the vertical column, opening additional windows may cause the system to override those changes. As subsequent opportunities occur for re-satisfying the user-specified constraints, the window management system attempts to do so.

The ViewPoint overlapping window system provides the user with controls for moving a window, resizing a window (both width and height), and placing a window

[1] Note that systems which maximize neither requirement are unusable, and those which maximize both must be prescient to a degree unattainable by current techniques.

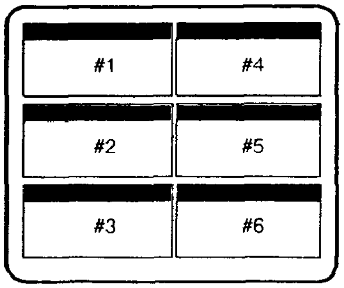

Figure 3. Default tiled locations.

on the top or bottom of other windows. Initially a window opens as a full page with a location offset from the last default window opening as shown in Figure 4. If the user chooses to manipulate the size and location of the window, the system remembers so that subsequently closing and reopening the window does not affect its position or shape.

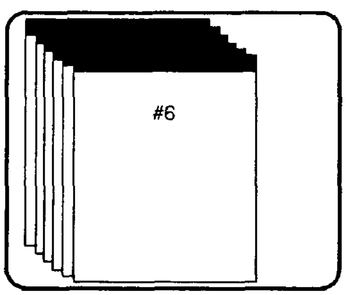

Figure 4. Default overlapping locations.

Subjects. There were 22 subjects, co-workers of the authors at Xerox Office Systems Division consisting of programmers, secretaries, and project planners, in their twenties, thirties, and forties. Thirteen were female. All had used ViewPoint to some degree, and each subject had at least a minimum of experience in at least one tiled or overlapping multi-window system. Four other subjects performed the experimental task but are not included in the final data; two in the Tiled-Irregular condition became too frustrated to finish the task, and two others were accidentally given misleading or incorrect instructions.

Tasks. The basic form of the experimental task was to look through four ViewPoint document files containing text and graphics, matching up graphic objects with brief paragraphs describing them. Only searching was required; the subject's answers were written on a piece of paper. Two variations of the task were created. One, called the Regular task, had the text and graphic objects arranged as rectangular regions in the upper left of the document pages (Figure 5). This task satisfied the predicted conditions for an optimal tiled task. The subject could view the entire content of these documents in four tiled windows (two on the left of the screen and two on the right). In fact, almost all of the text and graphic objects could be seen in the default window locations and sizes (three on the left and one full page on the right). Thus to solve that task with ViewPoint tiled windows required only one window operation (moving one of the windows to the right of the screen or scrolling one window slightly).

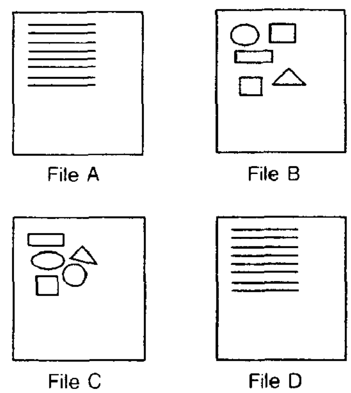

Figure 5. Files for the Regular task.

The other variation of the task, called the Irregular task, had the text and graphic objects arranged in an irregular manner within the documents (Figure 6). This task satisfied the predicted conditions for an optimal overlapping task. The contents did not conform to the tiled window sizes and shapes; a subject could see all

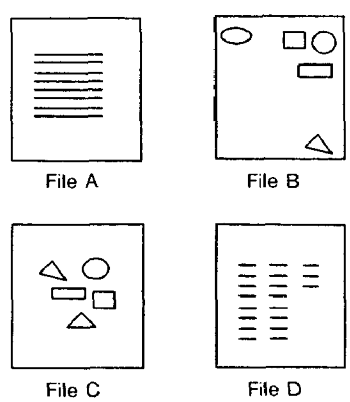

Figure 6. Files for the Irregular task.

the information at once only if the windows overlapped. The most efficient method of solving the Irregular task with tiled windows required a moderate amount of scrolling, or opening and closing different windows. Our prediction was that users working with tiled windows would take less time to solve the Regular task (because the system would take care of managing the windows), but more time to solve the Irregular task (because tiled windows could not accomodate the irregular contents as well).

Procedure. Subjects were tested individually in one of four conditions: using either tiled or overlapping windows and performing either the Regular or Irregular task.[2] The subjects were told that the experiment was to observe people solving a simple task using ViewPoint. The experimenters first reviewed various commands that the subjects might use, including the window manipulation commands. After indicating that they understood the task instructions, subjects performed the task and were timed with stopwatches by both experimenters. While the subject performed the task, the experimenters noted the frequency of the various window operations, and any errors. Afterwards, information was collected about the frequency with which the subject used the various tiled and overlapping window systems available at Xerox.

Results. Individual subjects' times to complete the tasks are plotted in Figure 7. As can be seen, even though there was a moderate amount of variability among subjects performing the Regular task, the median time of the group using overlapping windows was 30% longer (249 sec. vs. 185, $p < .05$ using the Wilcoxon test) than that using tiled windows.

In the case of the much harder Irregular task, subjects exhibited much more variability. In fact, the distribution for the subjects using overlapping windows is bimodal. We have called these two groups the Fast and Slow groups. The Fast group took about one-third less time to perform the task than subjects using tiled windows (274 sec. vs. 419, $p < .05$ using the Wilcoxon test). The Slow group took about one-third *more* time (546 sec. vs. 419, $p < .05$ using the Wilcoxon test). All but one of the Fast group were programmers thoroughly versed in the use of overlapping windows, and all but one of the Slow group were non-programmers less familiar with overlapping windows. Thus, it appears that the level of expertise distinguishes the two groups.[3] The wide range of variability suggests that such a distinction may also occur among the subjects using tiled windows.[4] We did not observe such a distinction between fast and slow groups in the two Regular task conditions, although that may be due to the small samples. In any case, any such differences there must be much smaller.

The number of window operations for the four conditions are given in Figure 8. As can be seen, the distributions are roughly the same as those for task-completion times. (The two measures have an overall correlation of .867, $p < .001$, although there are no reliable correlations for the individual conditions, possibly because of the small sample size.) Subjects using overlapping windows to solve the Regular task performed roughly twice as many window operations as those using tiled windows. Again, the results were less clear for the Irregular task. There was a difference between Fast and Slow users of overlapping windows (the four fastest subjects used the fewest operations). In contrast, the number of window operations used by the Tiled-Irregular subjects spanned the entire range used by both groups in the Overlapping-Irregular condition.

Since one could use overlapping windows to arrange the documents so that all the content was visible in the Irregular task, one would expect users of tiled windows always to perform more operations than overlapping-window users. The fact that one of the tiled subjects used fewer window operations than any of the overlapping subjects gives more support to the hypothesis that there were also "expert" and "inexpert" users of tiled windows. In fact, the three Tiled-Irregular subjects who were fastest also used the fewest operations in that condition.

Discussion. These results confirm that tasks which require little window manipulation can be carried out more quickly using tiled windows, and that tasks which require much more window manipulation can be carried out more quickly with overlapping windows. This general conclusion must be qualified, however, by consideration of user expertise. If we look just at users who are "inexpert" with overlapping windows, it is clear that tiled windows are better for both kinds of tasks; the advantage of overlapping windows is only available with a fair degree of practice. Furthermore, if expertise is important in the use of tiled windows as well (as suggested by the data on window operations) then expert users of tiled windows may be able to solve the Irregular task as fast as expert users of overlapping windows. This could mean that, for a given level of expertise, tiled windows are better in both situations. Though we would not want to draw such a strong conclusion from such limited data, it is certainly worth further examination. We should also note that, paradoxically, all but three of our subjects preferred overlapping windows.

5. Conclusions

We characterized the basic user requirements for a multi-window system as 1) windows that can conform to their contents, and 2) a system that relieves the user of window management. We further noted that

---

2 We ran some subjects through both tasks, but differential transfer effects rendered the second observations unusable.

3 This expertise may be due to any of a number of factors. For example, programmers typically spend many more hours per day working with many overlapping windows in the Xerox Mesa programming environment.

4 Recall that two subjects in the Tiled-Irregular condition were discarded from the data as they were unable to complete the task.

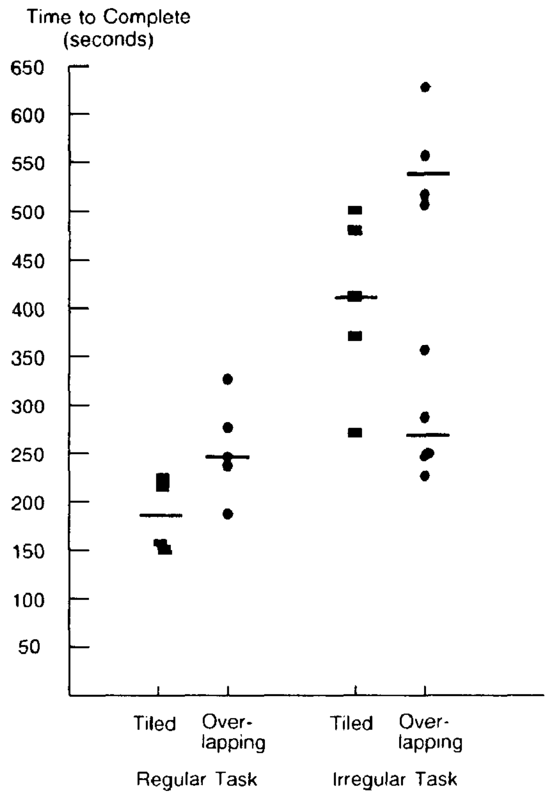

Figure 7. Individual and Median Completion Times for Each Condition.

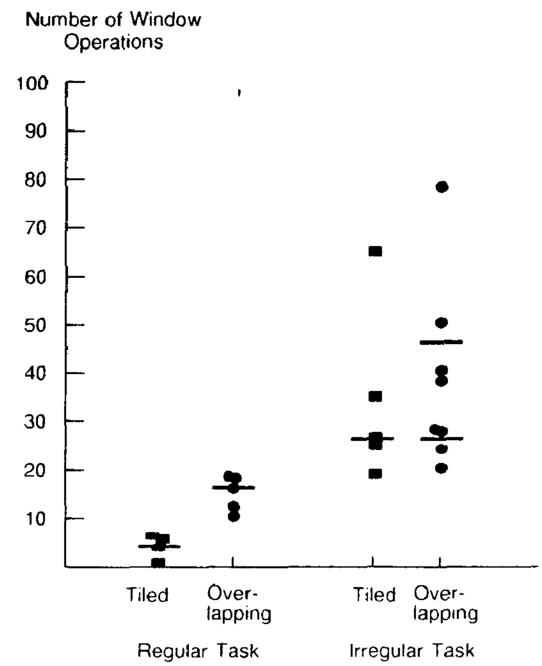

Figure 8. Individual and Median Number of Window Operations for Each Condition.

overlapping window systems emphasize the first of these characteristics, and tiled window systems the second. This analysis predicted that we should find situations where one or the other window management system would yield faster performance. The experimental results reported above support this analysis and its predictions However, our experiment looked only at tasks in which the window systems provided additional information and access to multiple sources of information, just two of the seven functions that Card *et al.* (1984) mention for multi-window systems. Furthermore, we examined only two of the many possible implementations of window management. Clearly this limited exploration of the design space is only a beginning. Our conclusion from this work is not to advocate either tiled or overlapping windows, but rather to emphasize the importance of further explorations. Effective window designs require more complete models of when and why a given window management system is preferable to another.

Acknowledgements. We thank Stuart Card for his comments in discussions. Rachel Rutherford for editorial suggestions. and our colleagues for being willing subjects.

References

Bury, K. F., S. E. Davies, and M. J. Darnell. (1985). Window Management: A Review of Issues and Some Results from User Testing. Technical Report HFC-53, IBM Santa Teresa Laboratory, Human Factors Center.

Card, S. K., M. Pavel, and J. E. Farrell. (1984). Window-based Computer Dialogues. *Proceedings of Interact '84: First IFIP Conference on Human-Computer Interaction*.

Cohen, E. S., E. T. Smith, and L. A. Iverson. (1985). Constraint-Based Tiled Windows. *Proceedings of the IEEE Computer Society Conference on Computer Workstations*.

Goodfellow, M. J. (1985). WHIM, The Window Handler and Input Manager. *Proceedings of the IEEE Computer Society Conference on Computer Workstations*.

Information Technology Center, Carnegie-Mellon University (1984). *User's Manual for Release 1 of the Information Technology Center Prototype Workstation*.

Lemmons, P. (1983a). A guided tour of VisiOn. *Byte*, 8(6).

Lemmons, P. (1983b). Microsoft Windows. *Byte*, 8(12).

Sheil, B. (1983). The Interlisp-D programming system. *Proceedings of CompCon Conference*.

Smith, D., E. Harslem, C. Irby, R. Kimball. (1982a). The Star user interface: an overview. *Proceedings of the 1982 National Computer Conference*.

Smith, D. C., C. Irby, R. Kimball, and W. L. Verplank. (1982b). Designing the Star User Interface. *Byte*, 7(4).

Teitelman, W. (1984). The Cedar Programming Environment: A Midterm Report and Examination. Technical Report CSL-83-11, Xerox Palo Alto Research Center.

Tesler, L. 1981. The Smalltalk environment. *Byte* 6(8).