GroupBar: The TaskBar Evolved

Greg Smith, Patrick Baudisch, George Robertson,
Mary Czerwinski, Brian Meyers, Daniel Robbins, and Donna Andrews
Microsoft Research
{gregsmi; baudisch; ggr; marycz; brianme; dcr}@microsoft.com

Abstract

Our studies have shown that as displays become larger, users leave more windows open for easy multitasking. A larger number of windows, however, may increase the time that users spend arranging and switching between tasks. We introduce GroupBar, a task management system for dealing with the profusion of windows on the PC desktop. Designed to offer the same basic form and function as the existing Microsoft Windows™ TaskBar, GroupBar additionally allows users to group windows into higher-level tasks and enables task switching with a single mouse click. In order to gain experience with GroupBar usage and to develop reasonable task definitions we conducted a longitudinal field study. Based on the results of that field study, we conducted a comparative user study wherein we found that participants were able to multitask faster when using GroupBar than when using the existing Windows TaskBar.

1. Introduction

Twenty years ago, Bannon et al. (1983) observed that information workers often switch between concurrent tasks. In Rooms, Card and Henderson (1987) observed that tasks can be supported by managing working sets of windows, in much the same way operating systems manage working sets in memory. They identified desirable properties of task management systems, including: fast task switching, fast task resumption, and easy reacquisition of mental task context.

Since then, many virtual desktop managers have been built and each exhibits some of these properties. Task management systems typically provide some efficient way of switching from one set of windows and applications to another set, as a basic form of task switching.

Although workers may switch among tasks in a self-guided manner, a significant portion of task switching is caused by external interruptions. Czerwinski, Cutrell, and Horvitz (Cutrell, 2001; Czerwinski, 2000; Czerwinski, 2000b) have sought to understand the influence of interruptions on task switching for information workers

in order to design user interface tools that can assist users to recover from interruptions.

We have also been motivated to re-examine task switching and task management design opportunities in the face of the growing popularity of larger display and multiple monitor configurations. In an informal study at our corporation, we found that when users shift to larger display surfaces, they leave more applications running and associated windows open. For example, we observed that single display users tend to keep an average or 4 windows open at once, while dual monitor users keep 12 and triple monitor users keep 18 windows open on average (N=16 users). Although a larger study is required for verification of these results, this significant trend suggests that there is an opportunity for design innovation with windows and task management to make dealing with larger numbers of concurrent windows a fundamentally more natural and effective experience.

We have developed a tool to exploit this opportunity. GroupBar is a desktop-resident toolbar, similar to the Microsoft Windows TaskBar, which allows users to arrange windows into groups and to switch between tasks with a single mouse click. The similarity to the Windows TaskBar was chosen as a design point in recognition of the fact that we could leverage user familiarity to reduce learning time, as well as to provide a basis for targeted comparison of task management features we developed.

In this paper, we will discuss related work, describe GroupBar, and present the results of a longitudinal field study of GroupBar, and a comparative user study of GroupBar and TaskBar.

2. Related work: task management

The most popular software system for task management is the virtual desktop manager. One of the earliest designs exploring a virtual desktop manager was Smalltalk Project Views (Goldberg, 1983). Rooms (Card, 1987; Henderson, 1987) is probably the most well-known of these kinds of systems. A number of these systems are currently available, and are described in (XDesk, 2003). We found that none of these systems have been evaluated in a stringent way; we could not find detailed reports on

studies demonstrating how easy to learn they are, or how well they integrate into real-world settings.

In addition to virtual desktop managers, a number of novel solutions have been proposed, including extending the user's desktop with additional low-resolution screen space (Baudisch, 2001), 3D solutions like the TaskGallery (Robertson, 2000) and Wurnig (1998), zoomable space as in Pad++ (Bederson, 1994), and the use of time as the main axis (Rekimoto, 1999). Also, tiled window managers (Bly, 1986; Teitelman, 1986) address some of these same issues, as well as systems that involve bumping other windows away (Bell, 2000; Kandogan, 1997).

We have pursued prototypes of temporal and spatial visualizations of users' daily computing configurations. These designs use lightweight, temporal cues, such as the state of a user's desktop at different times (Malone, 1983). We are also building support for task-based visualizations and switching, in a similar vein to the work of Henderson & Card (1987), Kaptelinin (2002), Macintyre et al. (2001) and Robertson et al. (2000).

In distinction to the prior work, we have sought designs for the virtual desktop organizers that don't replace the entire PC desktop with a new metaphor, but rather occupy the same conceptual and physical space as is already devoted to window management in the Windows OS – namely, the area along the edges of the display surface using the same minimized representations of top level windows. Using these prototypes, we are performing longitudinal studies on the benefits of temporal and visual cues for enhancing memory about knowledge-based tasks. We seek to understand the potential benefits from the use of these systems, and to iterate their design. For example, the Windows XP TaskBar provides "grouping by application" (executable) to address the problem of running out of space inside the bar. Grouping this way rather than by task has created confusion for many users, as specific application windows may not be related to each other, and cross-application windows might be related to the same task.

The ProjectBar, shown in Figure 1, was a first attempt to help users organize their documents and email by task (or "Project"), as opposed to by application. The default "Desktop" Project and three other user-defined Projects are shown. The currently selected Project, "CHI Paper", is expanded, so it shows its contents in the bar - i.e. the five window tiles. The other, inactive Projects are represented by clickable Project buttons. Hovering the mouse over the non-selected Project, "eMail", reveals a thumbnail fly-out image of what its desktop looks like.

The ProjectBar organizes open windows and applications so that associated items are grouped into a single Project. Projects are named and appear as a list of Project buttons in the bar, distinguished from the individual window buttons by color and shape. The list of Projects in the ProjectBar can provide a visual reminder of ongoing work and help with task management. When the user selects a Project by clicking on its button, the individual windows belonging to that Project are revealed on the desktop and the associated window buttons are revealed in the bar. In Figure 1, clicking on the "eMail[3]" button would hide the five window tiles belonging to the "CHI Paper" Project and reveal the three tiles belonging to the "eMail" Project. When users return to a Project, windows re-appear as they had been laid out previously on the desktop, preserving spatial context for the user as well as the resources needed for continuing the task.

We performed a pilot user study comparing ProjectBar with the Windows XP TaskBar to examine the ProjectBar's ease of use for multitasking with knowledge work. ProjectBar was found to provide higher satisfaction than the TaskBar.

Despite these encouraging results, we were ultimately not satisfied with how much user effort was required to learn to use the ProjectBar. Like many other systems, the ProjectBar requires users to explicitly create a task or virtual desktop before windows can be assigned to it. An alternative, offered on some virtual desktop switchers, is a fixed set of empty, pre-configured tasks or desktops that can be populated. The creation of new custom-tailored tasks remains a bottleneck.

We therefore built a lighter-weight version of the ProjectBar, which we call GroupBar. Our goal in designing GroupBar was to lower the bar to task creation and maintenance by allowing users to create or remove a task on the fly using a single dragging gesture.

The other issue GroupBar addresses is the accessibility of windows belonging to different tasks. While virtual desktop managers tend to have a strict separation between tasks, GroupBar allows users to simultaneously display any subset of windows, even if they should be assigned to different tasks.
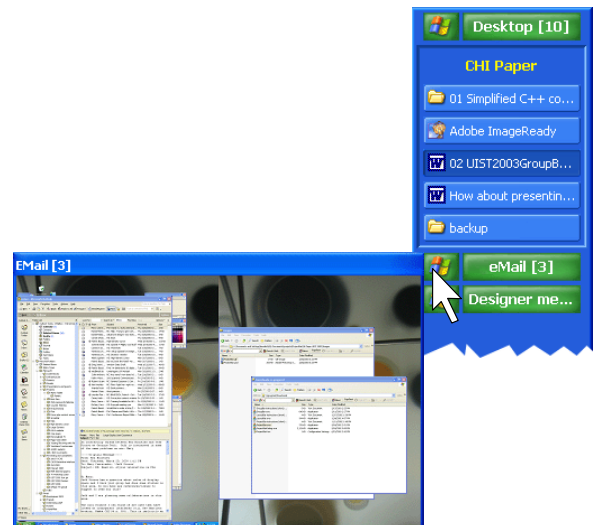

Figure 1: ProjectBar with four projects. As the user hovers the mouse over one of the projects, a fly-out provides a preview of that project.

3. GroupBar walkthrough

In this section we will describe GroupBar, how it allows users to interactively rearrange tiles representing windows, how it allows users to organize window tiles into high-level tasks called 'Groups', and how GroupBar allows users to switch between these tasks with the mouse.

Figure 2 shows two pictures of GroupBar, here in its vertical form factor. As in the Windows TaskBar, GroupBar contains one tile for each open window in the system. The current 'active' window tile is shown in a darker, depressed-button state, and any tile can be clicked on to activate (or to minimize, if already active) the corresponding desktop window.
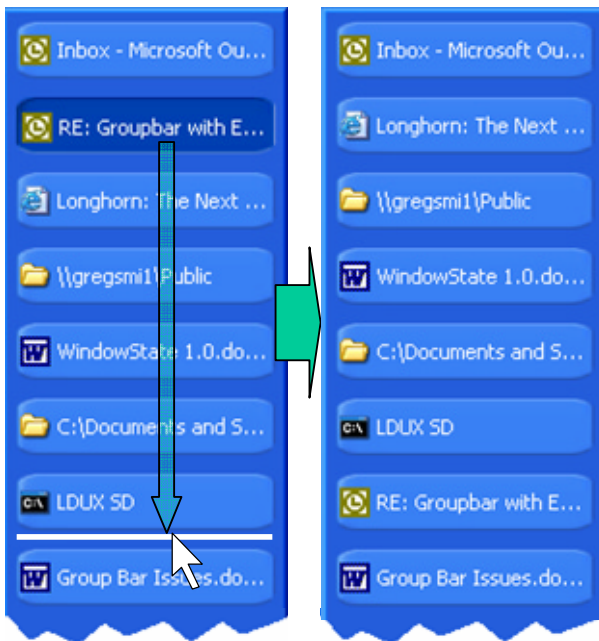

Figure 2: GroupBar allows users to rearrange tiles by dragging a tile between two other tiles.

The key enabling feature of GroupBar is the addition of drag-and-drop functionality to tiles on the bar. As the user drags a window tile along the bar, a white insertion caret appears, moving along the bar with the mouse cursor to indicate which of the possible drop locations will be activated when the tile is released (see also Figure 7).

The drag-and-drop functionality allows for two types of interactions. First, it allows users to *arrange* their window tiles freely, which may be used for giving tiles a more meaningful order or to establish a better correspondence between the position of a tile and the position of the represented window on the screen. Figure 2 shows the straight caret symbol that GroupBar uses to indicate that the dragged tile will be inserted at this location.

Second, users can drag a window tile *onto* another tile, which combines these two tiles into a Group. As shown in Figure 3, during the drag operation, a curved white caret (as distinct from the straight-line caret for

rearranging) is shown to indicate that the drop operation will result in Group formation. When a Group is formed, GroupBar visually unifies member tiles by surrounding them with a gray background and complementing the newly formed unit with a green "tab" at the top. Users can remove a tile by dragging that tile out of the group. Groups reduced to a single tile are automatically terminated and the group tab disappears.
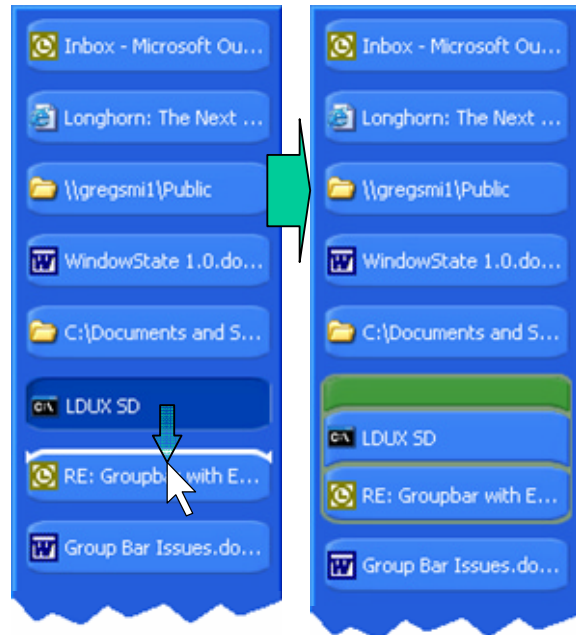

Figure 3: Dragging a window tile onto another tile combines both tiles into a "group".

The main point of GroupBar is that it allows users to perform operations on all of a Group's constituent windows simultaneously by manipulations of the Group tab. As further described in Section 4, the functionality GroupBar offers for Groups is directly derived from the functionality Windows offers for a single window tile. *Clicking* a Group tab restores all of the windows in the Group and brings them to the foreground. R*ight-clicking* the tab offers a context menu for additional Group actions (see Figure 4).

While the above figures depict a vertical bar style, GroupBar can be configured to either horizontal or vertical form factors, just like the standard Windows TaskBar (Figure 5). And like the standard TaskBar, GroupBar can be transformed from one form factor into the other anytime by dragging the bar to a different edge of the screen.

As we will show in the comparative user study below, GroupBar's ability to allow users to create, modify, and interact with entire groups of windows helps them increase their window management efficiency while requiring only a modest learning effort. Before we present our two user studies, however, we will discuss four of the design points in more detail.
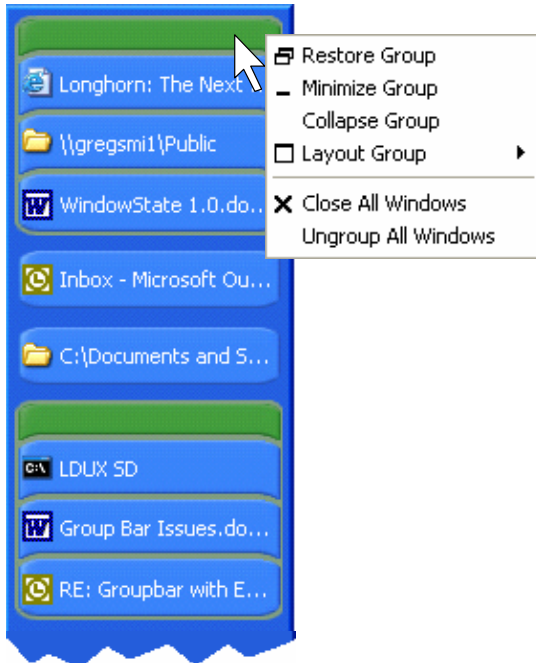
Figure 4: Clicking the green Group tab restores all of the windows in that Group and brings them to the foreground. Right-clicking a Group tab offers additional Group actions.

Figure 5: Fragment of GroupBar in its horizontal form factor.

4. GroupBar design

4.1. GroupBar is backwards compatible

The key addition GroupBar makes to the root interaction functionality of the existing Windows TaskBar is a drag interaction on the bar tiles. Since no drag interaction is defined on the tiles of the original TaskBar, users who choose not to use the grouping functionality can use GroupBar as if it were the regular TaskBar. When grouping is used, the additional grouping information is primarily conveyed via subtle changes in the shape and surrounding coloration of grouped window tiles, using the green Group tab as a high-contrast unifier. The size and visual distraction level of the Group tab is kept minimal by representing only the primary "switch to this Group" function directly in the interface; all other functions are relegated to a context menu. (Figure 6 shows an earlier design with an additional on-tab embedded button.)

One price to be paid for GroupBar's backward compatibility and unobtrusiveness to users who do not need grouping capabilities is that the grouping feature may

have low discoverability. We plan to address this issue by using two different approaches. One is to have bubble help appear in situations where grouping functionality might be useful, for example after a user has performed a series of different window minimization/restoration actions. The other possibility is to explore algorithms for automatic grouping based on analysis of the user's ongoing windowing activity, for example making an automatic Group out of a set of windows that seem to be highly co-incident on the screen. Since such automation runs the risk of being intrusive, it would have to be configured carefully. This feature was requested during our longitudinal field study.

Figure 6: Earlier GroupBar design: Without the tile curvature, the grouping is less apparent. This design features an additional "collapse" button for each Group.

4.2. Overloading drag interactions

GroupBar's drag interaction was derived from the drag interactions found in Windows file management and the toolbar customization functions of Microsoft Office. Porting the drag interaction was generally straightforward with a few special adaptations. Since a dragged window tile can be both appended to a tile or Group, and also inserted between tiles or Groups, the GroupBar drag and drop design had to simultaneously support grouping and re-ordering semantics.

As mentioned in the walkthrough section, GroupBar accomplished this by distinguishing dropping into Groups from dropping outside of a Group by altering the shape and location of the caret. Namely, Group insertion carets are curved and placed up against the tile with which grouping will occur, while non-Group rearrangement carets are straight and centered in the gap between two tiles. Figure 7 shows how the different caret shapes visually distinguish the different operations even though drop targets are closely co-located. Enabling these multiple carets was one of the main reasons for using the curved window tile design (compare to the non-curved design shown in Figure 6).

To help users acquire the individual drop target areas, we chose to decouple the target surfaces from the visual location of the caret symbols. As Figure 7 shows, the screen space containing the straight-line caret is too small to allow users to acquire this space efficiently. GroupBar solves this problem by distributing the screen space between tile centers evenly among the three adjacent targets independent of where the caret symbols appear. Our first experience with this assignment of target surfaces is positive and the benefit of larger minimum

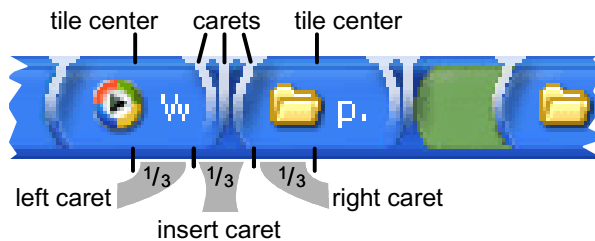drop areas seems to outweigh the lack of absolute positional precision.

Figure 7: Bar fragment and all possible carets over it. In order to allow users to easily acquire the drop position for insertion, GroupBar distributes the activation surface evenly over the various possibilities.
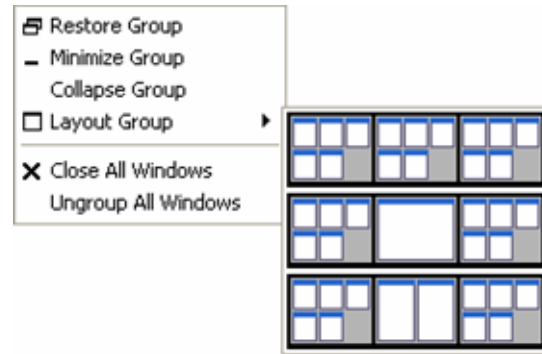

Figure 8: GroupBar context menu allows users to arrange all windows in that Group according to predefined layouts. Here the user uses a triple-monitor display, thus every layout extends across three screens.

4.3. Maximizing groups: window layout

The philosophy behind GroupBar is to allow users to operate on groups of windows as though they were a single unit. In existing Windows design, the left-click behavior of a tile button changes depending on the state of the window. For example, when a window is in a minimized state "Restore" is the most likely option, and "Minimize" is most likely when already restored. This is designed to optimize the button's use and allow invocation of several different functions serially through a single control surface. For familiarity and efficiency, we brought the analogous facility forward into the Group tab: If the Group is all minimized, left-clicking the Group tab restores all the windows and brings them forward. If the Group is already all restored, left-clicking the Group tab minimizes all the windows.

One operation that does not translate immediately from individual window tiles to Groups is the "Maximize" function. Maximizing each individual window in a Group would make them all overlap one another, which is likely to be more problematic than useful. GroupBar therefore extends the analogy with a twist by creating a "Layout Group" operation which serves to maximize the *collective* space taken up by the Group, rather than maximizing the space taken up by any individual window. Since there are several ways to accomplish this type of cooperative maximization, GroupBar currently allows users to choose from a selection of predefined layouts (Figure 8), but we recognize that users might like to define their own spatial layouts.

4.4. Handling large numbers of windows

As the number of displayed windows increases, any type of bar interface will eventually run out of space. The original Windows TaskBar deals with the issue by making users page through sets of tiles using small arrow handles (Figure 9a). Since this approach makes a large number of potentially relevant tiles difficult to access, GroupBar implements a different approach that leverages its knowledge about Groups. If GroupBar runs out of space, Groups can collapse into a more space-efficient representation in order to make room and prevent the bar from overflowing (Figures 9b and 9c). Groups to be collapsed are picked on a least-recently-used basis. We believe this will almost always be preferable over the "agglomeration by application" strategy implemented in the current version of Windows.

In order to allow users to use collapsing functionality to reduce visual clutter even before the bar overflows, users can manually collapse and expand Groups from the Group context menu.

As an alternative strategy, GroupBar users can deal with large numbers of windows by creating additional bar instances. Additional bars are added using the "Add New Bar" command from the GroupBar context menu. New bars are created empty. The user can position the newly created bar on any edge of any monitor, and then populate it using the already described drag-and-drop mechanism. By allowing users to place window tiles and Groups at widely separated screen areas, additional bars allow users to leverage spatial memory more heavily.
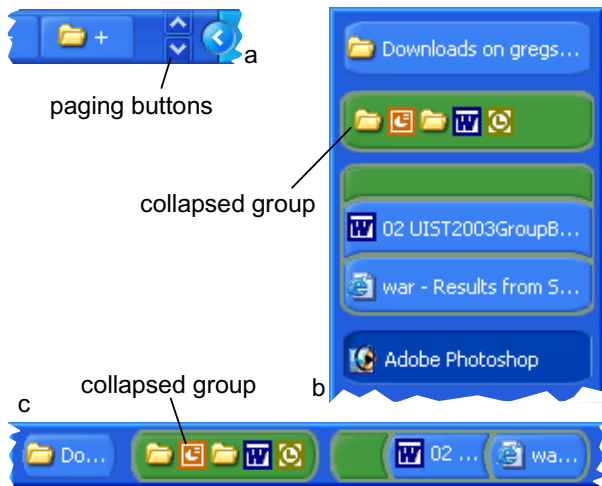
Figure 9: (a) Taskbar overflow vs. (b, c) Collapsed Groups in GroupBar.

5. Longitudinal user study

In order to gather information concerning how people actually use virtual desktop managers, and to begin to understand specifically how GroupBar might be used in real situations, we performed a longitudinal field study on a small number of subjects over a 7-10 day period.

5.1. Method

Five participants, aged between 20 and 60 and all male, were recruited to participate in the study. The participants were screened from a large local database of volunteers based on the criteria that they were experienced Windows and Office users. The users were also recruited to match design "personas" developed to represent target user groups. Space does not permit including the detailed specifications of each persona, but, broadly, the three personas used in the study included: 1 analyst, knowledge worker and light developer, 2 consultants, both knowledge workers, and 2 IT professionals. As may be obvious from the descriptions, all participants were fairly technology savvy.

A field study methodology was utilized in order to examine the usefulness and usability of GroupBar compared to the existing TaskBar. We used an *in situ* method to study the use of dual monitors and GroupBar in order to establish how important the new GroupBar features were for multiple monitors *using the participants' own work information and habits*. If, after approximately one week of use, participants were using the grouping features of GroupBar with their larger display space, this would provide evidence of the system's usefulness.

The five participants were first visited in order to collect baseline measures before starting the study proper. The baseline metrics included a measurement of how many windows users kept open with the TaskBar, inter-

view details about how they used their TaskBar and monitor for their work, and a survey about TaskBar features and relevant patterns of window management behaviors.

After the baseline visit, the 5 users were provided with the GroupBar executable and a very brief email tutorial on how to use its grouping features. Participants agreed to try GroupBar out as their main taskbar for 1 week. After using the new bar, they were visited in order to collect details and comments about how they utilized GroupBar, and how useful they found it to be. We also interviewed them about the kinds of tasks they chose to leverage GroupBar for, with an eye toward informing our design of the lab study, also presented in this paper, using realistic tasks.

5.2. Results

5.2.1. Baseline survey and usage data

Before using GroupBar, the participants reported typically running 5.6 applications or programs at once, on average, and that they kept an average of 6.4 windows open at once. All users reported using the original Task-Bar in the horizontal position. Two of the participants used Auto-hide, Always-on-top, Group-by-application and the "Close All Windows" feature available when Group-by-application is turned on. Only one participant reported using the "Tile windows vertically/horizontally" function in the TaskBar, and no one reported using the "Cascade windows" function. Only one of the participants knew that right-clicking on a window tile allowed them to close that individual window. This background data shows a fairly standard usage of the TaskBar, with users typically only using the minimize/maximize and close capabilities related to the TaskBar tiles.

5.2.2. Usage and user satisfaction with GroupBar

Once it was installed, all users were able to easily learn how to use the grouping features in GroupBar. We received no emails or calls for assistance after sending the installation instructions. In addition, users were able to easily integrate it into their existing work practices, as evidenced by their comments and grouping habits. This is what we had hoped to observe, and was our initial design goal.

After using GroupBar for one week, four of the participants filled out a user satisfaction questionnaire about the perceived benefits of the system, and areas in need of improvement. (One user did not return his questionnaires.) Users on average organized 2.5 groups within GroupBar, and 2 windows in each group, on average. Three of the participants chose to run GroupBar vertically, and 1 ran it horizontally. The user satisfaction findings were reasonably favorable overall for a first iteration user study, and two of the participants stated they

would continue to use GroupBar after the study, despite its lack of integration with the real TaskBar. GroupBar scored above average in response to questions such as, "It is useful to be able to group the tiles on GroupBar by dragging them 'on top' of each other", "It is useful to be able to close/open a group of windows all at once", "It is useful to have GroupBar remember a layout for a group of windows, so that they open in the same layout as when I closed the group", and "GroupBar makes multiple monitors more useful". Users were not as positive about having non-group windows minimize on a group switch, or about running more than one GroupBar at a time. All of the average ratings are listed in Table 1.

Question	Average Response (1=Disagree, 5=Agree)
It is useful to be able to rearrange the tiles on GroupBar by dragging them into a different order	3.5
It is useful to be able to group the tiles on GroupBar by dragging them "on top" of each other.	3.75
It is useful to be able to open a group of windows all at once.	3.5
It is useful to have GroupBar remember a layout for a group of windows, so that they open in the same layout as when I closed the group.	3.5
It is useful to be able to close a group of windows all at once.	3.75
It is useful to have windows that are not in use minimize when I click on another group ("Minimize on Group Switch").	2.3
It is useful to be able to have more than one GroupBar running at a time.	2.0
I spend less time re-laying and re-sizing windows now that I have GroupBar.	3.0
GroupBar makes multiple monitors more useful.	3.25
GroupBar makes multiple monitors more pleasant to use.	3.0
I feel that the two monitors with the Group Bar provide me with enough desktop space.	2.75

Table 1. Average user satisfaction ratings for GroupBar.

We asked users what features of GroupBar most helped them manage their open windows. Here were the responses from the four participants that provided comments:

- Participant 1: Being able to open 3 windows for different features of Outlook as a group has been very helpful. Still trying to get used to having multiple internet windows open.
- Participant 2: Used it much like I use the task bar to select the window I want to use.
- Participant 3: Being able to open, and close, a group of windows at the same time. When I restore a grouped set of windows, it remembers the positions of the windows.
- Participant 4: I can think of at least 3 sets of groups I would like to be able to launch on a single click. When they do come up, I want them grouped already.

Next we asked what more we could do to design better windows/task management support into GroupBar. Suggestions from the same four participants that provided comments are included below:

- Participant 1: It would be nice to have the start button and shortcuts on the group bar and eliminate the Taskbar altogether.
- Participant 2: It would also be nice to be able to right click close a window from GroupBar. *(Note: We have since added this capability into GroupBar.)*
- Participant 3: When a new window spawns from a grouped application, group it with the others. For example, when I open an email from Outlook it doesn't group with Outlook. I have the same issue with IE. A new browser window does not group with IE. I don't like having two bars (the TaskBar and GroupBar). Combine the capability of the Group bar into the Task Bar. I would like to stretch the TaskBar across the bottom of both monitors.
- Participant 4: Give GroupBar the smarts to auto-group icons that are commonly grouped together on each person's desktop. Allow us to pre-configure sets of applications that should be grouped together by default. I would like to be able to click an icon on GroupBar (or whatever) and have it launch a set of grouped applications specific to a task or function.

5.3. Discussion

The user study provided initial evidence that some of GroupBar design decisions were deemed valuable by users. For a first iteration design, users were able to easily figure out how to use GroupBar and were using the grouping features for their common tasks. Given the lack of ethnographic data available to designers of task management systems, this is an encouraging finding. In addition, two users stated they were going to continue using GroupBar after the study. These same users also liked the fact that the windows' positions were preserved in

GroupBar, so this feature will be maintained in future iterations.

The study further indicated interesting design ideas for improving GroupBar. First, we knew that GroupBar should be integrated with the current TaskBar but we did not take the time to add the Start Menu, system tray, etc. to the initial prototype. Users clearly wanted this and future work will head in this direction. Second, one user requested that children windows should group with the parent window's group. Finally, at least one user suggested that we move in the direction of allowing Group-Bar to "auto-group" applications that are typically joined together, based on frequency or recency of use.

We decided to take the initial feedback from our small longitudinal study and gain a more robust understanding of GroupBar's hypothesized ease of use over the Task-Bar by performing a laboratory study. The tasks we observed users performing with GroupBar in the field informed our choice of tasks to utilize as part of the lab study's design. In addition, we included a much larger sample of users. We expected to see significant performance benefits when multitasking across various documents and applications when using GroupBar, as opposed to the TaskBar.

6. Comparative user study

In our laboratory investigation, we created 3 tasks consisting of between 2 and 3 documents each, matching what we saw on average in the field. The tasks consisted of a "Spreadsheet", a "Joke" and an "Image" task. The Spreadsheet task required participants to go to selected cells in an Excel spreadsheet as indicated by a Word document, copy the contents of the cell at that location (a 9 digit random number), and paste it both in the Word document and in another Excel spreadsheet. The Joke task required users to identify typographical errors in a list of jokes in a Word document, copy them and paste them and the page number on which they occurred in an Excel spreadsheet. The Image task required participants to modify images in PowerPoint and in Paint based on instructions in a Word document. Two isomorphic sets of each of these tasks were devised so that they were of approximately equal difficulty (e.g., the random numbers in the Excel spreadsheet were rearranged, as were the selected cells, both Joke documents consisted of 2300 characters and had 23 typographical errors but were different jokes, and the instructions for how to modify each image simply asked the user to use different simple shapes or different colors).

6.1. Method

Eighteen participants (half female), all multiple monitor users and very experienced MS Office users as identified by a validated screener, were recruited for this study. Participants were 34 years old, on average, had used the computer for an average of 14 years, and said they multi-tasked between an average of 6 tasks.

Participants were given instructions about the overall study procedure and then allowed to read a brief overview of how each tool, the TaskBar or GroupBar, worked before proceeding. All participants were very familiar with the TaskBar and knew of most of its features, even if they didn't choose to use them. In order to ensure that they learned how to use GroupBar, users were guided through the grouping and layout of the documents that formed each of the three tasks for the study. In the TaskBar, participants could arrange the items in the TaskBar by application (as supported in the software) but not by task. However, we did allow them to lay out and size their task windows in a way that was best suited to each task before beginning. The TaskBar and GroupBar were both laid out vertically along the left-most bezel of the left-most monitor. While running GroupBar, the TaskBar was laid out horizontally and put on "auto-hide", so that users never saw it and interacted solely with GroupBar. Finally, the "agglomeration by application" mechanism in the TaskBar was turned on so that window tiles of like applications were juxtaposed, but they did not collapse into a single tile menu of all application windows.

In order to ensure that participants had to task switch between tasks, the experimenter interrupted them at set places in their tasks. Five task switches were required in order to carry out all three tasks to completion, the first three of these guided by experimenter interruptions between the Spreadsheet task and the Joke task when participants were approximately ¼ of the way through each, and then again at approximately ¾ of the way through the first task (Spreadsheet). After the first 3 interruptions, participants were told to finish the Joke task that they were in, go back and finish the Spreadsheet task, and then switch to the Image task and carry it through to completion. A twenty minute deadline procedure was used for each of the three tasks to keep the session length under two hours.

The study was run on two identical, late model Compaq Evo machines with triple flat panel LCD monitors running at 3840 x 1024 resolution. Late model MS keyboards and the IntelliMouse were used for input. Windows and Office XP were the base OS and applications used in the study. The order of software tool used and task set were counterbalanced across participants. Participants were run in pairs each session.

Dependent measures collected included task time, subjective satisfaction responses to a questionnaire presented after using each tool, and overall tool preference. Task times were recorded using a countdown program on the participants' machines. A log of users' activities in terms of window management and group interaction was collected; analysis of that data is ongoing.

6.2. Results

The task times revealed a strong positive skewing, therefore a log transformation was applied as is standard to correct for a non-uniform response time distribution. A t-test of the log task times revealed a borderline significant task advantage for GroupBar, $t(34)=1.5$, $p=.07$, one-tailed. GroupBar average task time was 11.7 minutes, while the average for the TaskBar was 13.25. The task time data including one standard error of the mean in each direction are presented in Figure 9.
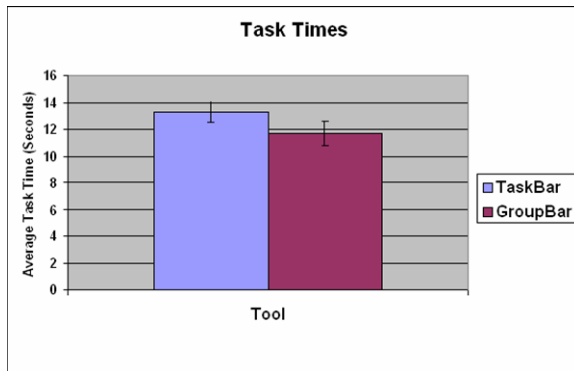

Figure 9: Average task times +/- one standard error of the mean for TaskBar and GroupBar.

Survey Question (1=Disagree, 5=Agree)	TaskBar	GroupBar
Task switching was easy to perform using the…	2.95	4.63
It was hard to go back and forth between my various windows and applications us-ing…..	3.32	1.42
I was satisfied with the functionality of the ….	2.68	4.42
The TaskBar/GroupBar is an attractive innovation for Windows.	3.16	4.42

Table 1: Average satisfaction ratings for the TaskBar and GroupBar. All ratings were significantly in favour of GroupBar at the p<.05 level.

In terms of overall satisfaction with the software, participants strongly favored GroupBar over the TaskBar. Individual t-tests of each questionnaire item revealed satisfaction ratings significantly in favor of GroupBar on every question (an overall ANOVA was not possible as questionnaire items were worded both in the positive and in the negative in an effort to avoid presentation bias).

Strict confidence levels were adopted (Bonferroni corrections) to account for multiple tests but all p-values were well below the .001 level. Table 1 summarizes the average ratings for each question.

Finally, GroupBar was unanimously preferred over the TaskBar. Despite this, many participants suggested improvements to GroupBar. Most frequent requests were for color coding or labeling of the groups in the bar, and for tooltips for document names when the group is collapsed. These features can be easily added to GroupBar. Several expert users wanted to see better keyboard accelerators like ALT + TAB enabled in GroupBar as well.

6.3. Discussion

While the field study suggested to us that GroupBar was discoverable and considered valuable by the participants, the laboratory study allowed us to better verify these benefits in a more controlled setting. Using tasks similar to what we observed users doing in the field, we had users switch back and forth between multiple tasks, prompted by interruptions or task completions. Group-Bar provided benefits over the TaskBar both in terms of overall task time (a borderline significant result) and in terms of users' perceived satisfaction with the task switching (strongly significant). Users commented that the tasks and interruptions forcing the switches were similar to what they experienced in the real world, so we feel we succeeded in simulating an information worker's daily task juggling. The study provides further evidence that software tools like GroupBar can provide user assistance as users manage multiple, complex tasks.

7. Conclusions

GroupBar provides basic task management, making it easy to group windows into a task using a straightforward drag and drop interaction. Task switching is accomplished with a single mouse click. Additional functionality on all windows in a task (*e.g.,* minimizing, restoring, and closing) has been determined through two studies to be easy to use.

The major difference between GroupBar and other task management systems lies in how the tasks are shown to the user. Rooms and other virtual desktop systems show all the tasks only when users go to an overview. GroupBar displays the tasks in a form that is a subtle extension to the familiar Windows TaskBar. This was shown to be a great advantage in terms of learnability and user acceptance, as demonstrated by our *in situ* and laboratory user studies of GroupBar.

Further user studies of GroupBar and its extensions will lead to design improvements and help clarify which designs are most appropriate across users and various display configurations. Competitive studies will clarify advantages and disadvantages of these two systems compared with existing systems. It is clear from our early studies on these topics that such new task management

systems are considered valuable by information workers, especially in multiple monitor configurations.

8. Acknowledgements

9. References

Baerbak Christensen, H. (1997). *A software development environment based on a geographic space metaphor.* Technical report, Univ. of Arhus.

Bannon, L., Cypher, A., Greenspan, S., and Monty, M. (1983). Evaluation and analysis of user's activity organization". In *Proc. CHI'83* (pp. 54-57). NY: ACM.

Baudisch, P., Good, N., Stewart, P. (2001). Focus plus context screens: combining display technology with visualization techniques. In *Proc UIST 2001* (pp. 31-40).

Bederson, B. & Hollan, J. (1994). Pad++: A zooming graphical interface for exploring alternative interface physics. In *Proc. UIST'94* (pp. 17-26).

Bell, B. and Feiner, S. (2000). Dynamic space management for user interfaces. In *Proc. UIST'00,* (pp. 238-248).

Bly, S., Rosenberg, J. (1986). A comparison of tiled and overlapping windows. In *Proc. CHI '86* (pp. 101-106).

Bury, K. F., Davies, S. E., and Darnell, M. J. (1985). Window management: A review of issues and some results from user testing. *IBM Human Factors Center Report HFC-53,* San Jose, CA.

Card, S.K. & Henderson, A.H. Jr. (1987). A multiple, virtual-workspace interface to support user task switching. In *Proc. CHI+GI 1987* (pp. 53-59). NY: ACM.

Cutrell, E., Czerwinski, M. & Horvitz, E. (2001). Notification, Disruption and Memory: Effects of Messaging Interruptions on Memory and Performance. In *Human-Computer Interaction--Interact '01* (pp. 263-269). IOS Press.

Czerwinski, M., Cutrell, E. & Horvitz, E. (2000). Instant Messaging and Interruption: Influence of Task Type on Performance. In *Proc. OZCHI 2000 (pp. 356-361).* Sydney, Australia.

Czerwinski, M., Cutrell, E. & Horvitz, E. (2000b). Instant Messaging: Effects of Relevance and Time. *Proc. HCI 2000, Vol. 2,* (pp. 71-76). British Computer Society.

Czerwinski, M. & Horvitz, E. (2002). Memory for Daily Computing Events. In *Proc. HCI 2002,* (pp. 230-245).

Goldberg, A. (1983). *Smalltalk-80.* NY: Addison-Wesley.

Grudin, J. (2001). Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proc. CHI'01,* (pp. 458-465).

Guimbretiere, F., Stone, M., and Winograd, T. (2001). Fluid interaction with high-resolution wall-size displays. In *Proc.UIST'01,* (pp. 21-30). NY: ACM.

Henderson, D. A. Jr., Card, S.K. (1987). Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics, 5* (3), 211-243.

Kandogan, E. and Shneiderman, B. (1997). Elastic Windows: evaluation of multi-window operations. In *Proc. CHI 97.* (pp. 250-257). NY:ACM.

Kaptelinin, V. (2002). UMEA: User-monitoring environment for activities. In *Proc. UIST'02 Companion,* (pp. 31-32).

MacIntyre, B., Mynatt, E., Voida, S., Hansen, K., Tullio, J., Corso, G. (2001). Support for multitasking and background awareness using interactive peripheral displays. In *Proc. UIST 2001,* (pp. 41-50).

Malone, T. W. (1983). How do people organize their desks? Implications for the design of office automation systems, *ACM Transactions on Office Information Systems 1* (1), 99-112.

Myers, B. (1988). Window interfaces: A taxonomy of window manager user interfaces, *IEEE Computer Graphics and Applications, 8* (5), 65-84.

Mynatt, E., Igarashi, T., Edwards, W., and LaMarca, A. (1999). Flatland: new dimensions in office whiteboards. In *Proc. CHI'99,* (pp. 346-353).

Patton, B.M. (1990). The history of memory arts. *Neurology, 40,* 346-352.

Rekimoto, J. (1999). Time-machine computing: A time-centric approach for the information environment. In *Proc UIST'99* (pp. 45-54).

Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D., and van Dantzich, M. (1998). Data Mountain: Using spatial memory for document management. In *Proc. UIST'98* (pp. 153-162).

Robertson, G. van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D., Gorokhovsky, V. (2000). The task gallery: a 3D window manager. In *Proc CHI'00* (pp. 494-501).

Teitelman, W. (1986).Ten years of window system – A retrospective view. In Hopgood, F., Duce, D., Fielding, E., Robinson, K., & Williams, A. (Eds.). *Methodology of Window Management.* Berlin: Springer-Verlag.

Wurnig, H. (1998). Design of a collaborative multi user desktop system for augmented reality, in *Proc. Central European Seminar on Computer Graphics*.

XDesk Software (2003), About Virtual Desktop Managers, http://www.virtual-desktop.info.