

Revisiting Display Space Management: Understanding Current Practice to Inform Next-generation Design

Dugald Ralph Hutchings

John Stasko

GVU Center/College of Computing
Georgia Institute of Technology

Abstract

Most modern computer systems allow the user to control the space allocated to interfaces through a window system. While much of the understanding of how people interact with windows may be regarded as well-known, there are very few reports of documented window management practices. Recent work on larger display spaces indicates that multiple monitor use is becoming more commonplace, and that users are experiencing a variety of usability issues with their window systems. The lack of understanding of how people generally interact with windows implies that future design and evaluation of window managers may not address emerging user needs and display systems. Thus we present a study of people using a variety of window managers and display configurations to illustrate manager- and display-independent space management issues. We illustrate several issues with space management, and each issue includes discussion of the implications of both evaluations and design directions for future window managers. We also present a classification of users' space management styles and relationships to window system types.

Key words: display space management, window management, multiple monitors, interview

1 Introduction

The personal computer (PC) has undergone dramatic changes over the past 25 years. The huge gains in processor speed and physical memory size and the popularity of the Internet have allowed people to use computers in an astonishing variety of ways. Throughout this period, the desktop and related window system have changed very little. Many organizations and individual enthusiasts have built window managers and interaction techniques to help users allocate space to windows and information, but few have been formally evaluated. Furthermore, because of a very small number of studies of the actual practices window manager users (which for PCs is nearly *all* users), we know very little of the details of specific managers and

of general principles of window management.

The latter aspect is more crucial as we experience the rise of more complex display environments. The increased abilities of video cards and the decreasing cost of display devices are potential indicators that multiple monitor systems will become more common; Czerwinski *et. al.* have already shown that in the presence of multiple monitors, task completion time is improved [6]. Grudin exposed fundamental differences between multiple monitors and larger displays and demonstrated how users interact with multiple monitor systems much differently than with single monitor systems [9]. Improvements to window management for emerging display systems could benefit from a better understanding of display space use characteristics.

In this paper we present results from a study of 20 users of various display configurations and window managers. After discussing related work, study method, and some definitions, we give a classification of our participants' management styles. The majority of this paper examines the general management principles discovered in our interviews with study participants, with implications for design and evaluation shown for each principle. Finally, we suggest possible directions for future work in space management.

2 Related Work

Relatively little research on understanding people's space management practices has been performed. Grudin studied multiple monitor users to understand the unique characteristics of such systems, finding some surprising uses and associated usability problems [9]. Ringel studied how people used virtual desktops, and compared findings to those of multiple monitor users [17]. Mynatt examined office workers' use of physical whiteboards to inform the design of Flatland, an electronic whiteboard system [16]. Each of these researchers employed structured interviews of smaller user populations (14 to 20 people), thereby providing detailed, focused studies.

Other work has focused on specific characteristics of window management. Gaylin videotaped nine users

(eight programmers) of an overlapping window system and logged the window operations they performed. He used the results to build better benchmark tasks for testing overlapping window managers [8]. Hutchings *et al.* more recently used this technique to compare single-monitor and multiple-monitor users [12], finding for example that email windows are used differently by each user group. Card *et al.* [5] present the “window working set” model (which inspired the Rooms system [10]) and a classification of window functions, which Bly & Rosenberg used to find their tiling scheme superior when information in each window appeared in the same location [4]. Kandogan & Shneiderman later found their Elastic Windows tiling scheme superior for a number of specific tasks [13]. Neither set of tiling experiments tested tiling schemes against more sophisticated overlapping systems such as Rooms.

Many systems have been proposed to help users with various aspects of window management, but lack formal user testing. Rooms is an early, comprehensive virtual desktop system, allowing users to arrange sets of windows (to correspond with tasks) and switch among the sets with very simple input actions [10]. SCWM also allows users to group windows, but the groups are formed by using a constraint solver [1]. Peeling and Rotating Windows aim to let users more easily access different windows by treating them more as physical pieces of paper and less as rigid rectangles [2]. Non-overlapping Dragging allows users to reposition windows anywhere and automatically moves any resulting obscured windows to nearby empty spaces, helping to keep windows visible [3]. QuickSpace also uses empty space to grow one window while keeping other windows visible [11]. CIWM relieves the user of creating, deleting, moving, and resizing windows through the use of artificial intelligence [7]. The Task Gallery, a 3D task and window manager, is a notable exception to these systems and techniques, providing several user studies of the system and showing how easily users could switch among different tasks [18]. In the commercial arena, Mac OS X Exposé includes two operations for finding windows in a cluttered desktop by providing a scaled overview of open windows [14].

3 Methods

We interviewed 20 adults (11 female) in their regular workspaces. Each interview ranged between 30 and 60 minutes. With participant permission, the interviewer tape-recorded the interview, captured screen contents at various points, and photographed the physical work environment. The interviews were structured so that the interviewer asked each of the participants the same questions from a written script. The interviewer had the flexibility to ask follow-up questions when participant responses warranted further investigation. Some of the

questions asked the participants to demonstrate how they interact with windows in different situations, and in some of the following sections we analyze those observations. The interviewer made few notes during the interviews so as to focus on window interaction. The audio recording was later carefully analyzed to more fully understand participants’ comments about their window use.

We recruited participants both from within our broad computer science department (excluding people in our own research group) and through contacts in outside organizations in order to have a population of different window systems and occupations. 17 people had at least one desktop PC, 12 of which were single-monitor, three of which were dual-monitor, and two users had two independent single-monitor systems in the same desk space. Three people used a laptop exclusively, one of whom connected the laptop to another monitor when he used the laptop on his desk. Thus there were 22 window systems: CDE on Solaris (3), Enlightenment on Linux (2), KDE on Linux (1), Mac OS 9 (2), Mac OS X (1), Microsoft (MS) Windows 2000 (7), and MS Windows XP (6). Five people employed virtual desktops. Eleven of the 20 participants were students: chemistry (2), computer science (5), general studies (1), immunology (2), and mathematics (1). Of the nine professionals, occupations included administrative assistant (3), computer science professor (2), system administrator (2), user interface designer (1), and virology researcher (1). Of the 20 participants, eight worked outside of the computer science department. Thirteen users identified themselves as *constant* users of their systems, three as *occasional* users (meaning that they used their system for one or two hours per day), and four as *fluctuating* users (some days they are constant users and other days they are occasional users). Screen resolutions ranged wildly, with the endpoints of 800×600 pixels and 3200×1200 pixels.

4 Definitions

Because we had participants with a variety of display configurations, we take care here to use different meanings for *monitor* and *screen*. *Monitor* means an independent display device whereas *screen* means the entirety of display area. A participant with multiple monitors or systems has only one screen.

We point the reader unfamiliar with windows to an excellent overview of window management by Myers [15]. Two terms that Myers does not specifically address are *maximize* and *stacking*. *Maximize* refers to growing a window to fill an entire monitor. *Stacking* refers to changing the depth (*z*-position) of a window.

Beyond basic definitions, it is important to carefully define the idea of *focus* (indeed, Myers briefly refers to the concept). We use *input focus* to refer to the window

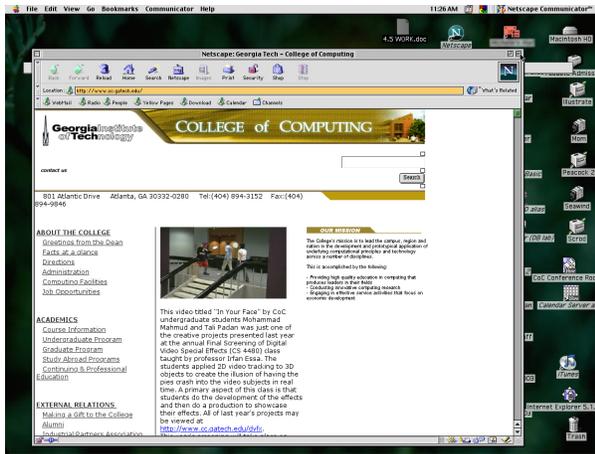


Figure 1. A screen of a near maximizer. Windows are as large as possible without covering important icons.

that has system focus, *i.e.*, the window that exclusively receives input from the user. We use *user focus* to refer to the window that the user is actively viewing, which may or may not have input focus. Given that tasks often involve multiple windows, the separation is crucial since completing a task can involve many changes in user focus, but few changes in input focus. When we use *focus* without qualification, we mean both input focus and user focus.

5 Management Styles

We witnessed four dominant ways that participants switched among windows: (1) moving the mouse directly to the window, sometimes requiring a click to stack it, (2) using a keyboard sequence (usually $\langle \text{alt} \rangle + \langle \text{tab} \rangle$), (3) using a special, fixed interaction area (*e.g.* MS Windows' *taskbar* or Enlightenment's *dock*), and (4) iconifying a topmost window to stack the desired window. Most participants indicated that they exclusively used one technique, although a few mixed techniques (mostly mixing (1) with (3)). Among virtual desktop users, participants used methods (2) and (3) to switch among desktops.

Participants had a variety of ways in which they organized screen space; no two organized windows in *precisely* the same way. However, participants fell into three broad categories: maximizers (five participants), near maximizers (5), and careful coordinators (10).

Maximizers simply maximize every window (or almost all windows). All maximizers used a taskbar or $\langle \text{alt} \rangle + \langle \text{tab} \rangle$ to switch among windows. This outcome is expected as the taskbar and keyboard can be easily accessed regardless of window size. All maximizers were single-monitor users, all but one had screen resolution of 1024×768 pixels (the other had 800×600 pixels), and all used MS Windows even though other

systems (such as Solaris CDE) include the *maximize* window operation.

Near maximizers (NMs) are slightly but importantly different. These participants had one or more smaller windows with which they frequently interact or glance (such as instant message (IM) clients or computer status indicators) *or* leave a bank of desktop icons uncovered. Users will manually resize (nearly) all other windows to occupy all but a little of the remaining portion of the monitor. Interestingly, no NMs used *always on top* window features, if such features were available. When asked about this, each replied in the following vein as Participant 9: “[Sometimes] I have to fully maximize a window, which means having an always on top window is annoying.” Figure 1 shows a screen shot of a NM. Every NM used mouse-direct switching to move between a nearly maximized window and a window or icon elsewhere in the screen. But NMs composed a variety of interaction techniques for switching among nearly maximized windows. Two MS Windows users used the taskbar, two Mac OS 9 users used iconify-restore cycles, and the one virtual desktop user kept one “nearly maximized” window per desktop.

Careful coordinators (CCs) are those who tended to have many windows visible simultaneously (meaning that none of them are maximized) or when they had a maximized window, were working in an application that itself had many sub-windows. We have dubbed this group as “careful” because most, if not all, visible windows had an important function for the user and were arranged to reflect that function. CCs also tended to have similar *widths* for similar applications (each web browser had the same width, each IM message window had the same width, etc.), although the length of each window might vary. All but one of the virtual desktop users who we interviewed were CCs. CCs seemed to use a taskbar only when the window to be switched to was not visible. All but one of the CCs had screen resolutions greater than 1024×768 pixels, and several CCs indicated that their monitors were too wide to use maximized windows.

6 Usage patterns in NMs and CCs

All of the participants indicated that everyday interaction involves coordination of multiple windows. Many factors contribute to the way that participants manage the many windows occupying the screen, completely irrespective of display configuration and system characteristics.

6.1 Invisibility is as important as visibility

Using information from one window to interact with another window is quite common, whether it be consulting an outline in order to write a paper, compiling email message comments into a coherent

digest, grabbing images from web pages, or using documentation to write a piece of computer code. Users employ moving, resizing, and stacking of windows in various ways to accommodate the visibility of information. But participants also use these techniques to purposefully hide information as well! We discuss two different reasons that people hide window contents.

One instance of purposeful hiding is when the interaction in one window (the *main* window) can be aided by information in other windows (the *secondary* windows). User focus will shift among the entire set of windows, while input focus will mostly remain in the main window. When the main window has user focus, secondary windows can be distracting for many reasons. Secondary window animation is *change-blind* if the change is so subtle that it will not grab the user's focus; a complaint voiced by six users is the presence of non-change-blind animation. However, a portion of the information area (as opposed to interaction area) can be important for users to view, making iconification of the window impossible. Thus, users will attempt to hide the distracting areas by moving them off-screen or by allowing the main window or secondary windows to cover them. Another distracting factor could simply be the sheer amount of information contained in a secondary window, relative to the information that is relevant to the main window. Thirteen users indicated that they hide a large portion of the secondary window(s) to allow them to more quickly locate the relevant information and focus on the task at hand. These participants often expressed that resizing secondary windows is undesirable as the layout of the information is then subject to change, causing disorientation and unnecessary interaction. For example Participant 7 often displayed the outline of a document in a separate window from the document itself to help guide the writing of the document.

Another type of information hiding relates to privacy, as indicated by six users. All participants used email and many had email programs running constantly. A number of participants also used IM to communicate. Some used programs with sensitive, proprietary information. This can pose a difficult management situation, since information is frequently accessed or consulted, yet should remain invisible when not in direct use. As Participant 2, who works in a laboratory, said, "The desktop is not as personal as just one person's vision," or, as participant 14, who has a private office, stated, "I don't want to have [my email] visible on the screen when people walk in. I'm pretty private about it... hiding things is good." A few users iconify communication client windows or place them on a dedicated virtual desktop to completely hide them. However, because these clients are frequently used,

most users partially hide the clients behind other windows. In particular, IM requires focus switches over short intervals, making iconification or placement to other desktops inefficient due to excessive interaction.

Implication for design: All participants who employed hiding techniques demonstrated situations in which hiding information was difficult or caused an unnecessary amount of interaction. A design that arises immediately from our observations is an operation that shows or hides a user-specified region of a window, whether for information display or maintaining privacy. One might also consider dynamic transparency or other subtle methods of obfuscation for privacy. Both are better suited for the window manager because an application designer will likely not know *a priori* what information will be displayed and valued by the user.

Implication for evaluation: This finding can help to create appropriate tasks for testing a display space manager: tasks may include the use of many windows as information-only windows or the need to hide part of a window and show another part of that window to successfully complete a task. An evaluator might specifically test the mechanics of showing or hiding parts of an individual window.

Implication for a specific system: This finding could have a positive impact on a system like SCWM [1], which lets users group windows through constraints. For example, a user could constrain a secondary window with sensitive information to always be occluded (when it does not have input focus) by the primary window in a multi-window task.

6.2 Strict tiling is rarely employed

A consequence of intentional hiding of information in windows is that participants rarely tile them. However, we found two additional direct causes for the absence of tiling strategies by our participants. Responses to two sets of interview questions helped us understand why. When the interviewer first arrived, he asked the participants to explain the layout of windows on the desktop. Later in the interview, the participant showed how he or she interacted with windows in a typical or recently-completed multi-window task. In all cases, participants did not have or use tiled windows.

One cause was the desire to quickly access many other windows, as all users indicated. In contrast to the times when one window dominates input focus and user focus is spread among many windows, there are other times when both input and user focus frequently switch. In these cases, leaving just a small bit of a window visible makes it easily accessible, and furthermore allows the window to maintain the layout of its information. Participants using multiple monitors in particular indicated that because special interaction areas like the taskbar are located on only one monitor, it

is easier to directly click on windows than to travel to the taskbar and then back to the window of interest (replicating similar results from Czerwinski *et. al.* [6] and Hutchings *et. al.* [12]).

The other cause was the desire to prefer the display of information rather than UI components in windows, especially when such windows did not have input focus. Consider the IM client in Figure 2, which was used by Participants 2 and 10. Each one situated the window such that the bottom left section (nine UI buttons) was off-screen. They did so because when they looked to see who was online, they did not need to see the buttons, and displaying the buttons wasted screen space that they could use for other purposes. We selected this example because it makes for a small figure and is indicative of six other participants in our study who demonstrated similar attributes with other windows.

Implication for design: A general disadvantage of tiling window managers is that carefully positioned windows are easily disturbed, which can cause the issues of accessing windows and wasting screen space. Possible design directions include exploring alternate ways of tiling (perhaps by tiling sections of windows and not entire windows) or methods of quickly reverting to previous configurations. In the case of multiple monitors, tiling systems must help users access windows and also respect physical monitor boundaries.

Implication for evaluation: A successful evaluation of a tiling system would demonstrate how well the system overcomes the issues of wasted screen space and extra interaction needed to quickly access a larger set of windows. Selecting tasks that require the use of a large number of windows will help. Additionally, tiling for multiple monitor users is not well understood and evaluation may lead to advances in design.

Implication for a specific system: Demonstrating that users indeed struggle to manage windows so that they can quickly access them justifies the utility of Rotating Windows [2], which is designed specifically to decrease the effort needed for this type of configuration.

6.3 “Empty space” is often not empty

One set of interview questions asked the participants to show the typical layout of windows on their systems. Three participants completely filled the screen with windows, leaving no pixel of screen space without a window. The 12 others left some screen space vacant, and all of them (except virtual desktop users) specifically managed the windows to keep a bank of desktop icons visible. In all cases, these desktop icons provided a way to create a new window and did not represent a window that was already created. Figure 1 shows a near maximizer’s icons; note that some careful coordinators also exhibited this behavior. Most participants indicated a desire for an easier way to keep

these shortcut icons visible. The surprising finding is not that participants used icons frequently, but rather that they have *specific* icons visible almost all of the time and that the visibility of these icons greatly affects the way that they manage space.

Participants demonstrated many functions that icons can serve (sometimes more than one simultaneously). Icons can act as “quick launches” for commonly used applications or files, “status monitors” for events such as print jobs, important interactive components (such as the “Trash” or “Recycle Bin” icon, which allows people to temporarily delete files), easily accessed temporary files, and even visual reminders to complete a task. The difference between reminders and “quick launches” or “temporaries” is that reminder files need urgent user attention, whereas the other two do not. Many participants indicated covering the former was an annoyance, covering the latter was detrimental to work. This situation causes participants to complete a lot of manual resizing and avoid automated window functions like *maximize*. They also avoided operations that hide all windows and show the desktop (available in both Windows XP and Mac OS X Exposé as keyboard shortcuts) because they would have to remember to use the operation to look at the icons on the desktop.

Implication for design: Future systems might explore how to designate a group of icons as “non-empty space,” where, for example, *maximize* does not cover the space, but manual resizing of windows allows the icons to be covered. Alternatively, the notion of desktop icons could be replaced by something that more tightly integrates with the window system.



Figure 2. On the left are the top and bottom sections of an instant messenger client. On the right is the middle section, which is the only useful part when it has user focus but not in input focus.



Figure 3: Participant 5 leaves four full windows visible reminding him to complete tasks later (outlined in white). The fifth outlined section at the bottom of the image is the dock, where three additional windows act as yet more reminders. Use of multiple monitors in particular seems to engender using windows as graphical reminders.

Implication for evaluation: Evaluation of emerging systems should include an assessment of how easily users can switch to *and interact with* the desktop, then revert to an original window configuration. Systems that replace the icons concept might gain from a comparison to a system that uses icons to assess both usability and learnability of the new system.

Implication for a specific system: Illustrating how users need “empty space” to remain “empty” indicates that designs such as Non-overlapping Dragging [3] and QuickSpace [11], which both try to exploit empty space by automatically moving windows, should proceed very carefully. The techniques could prove to be unusable if their obscure off-displayed icons.

6.4 Windows can act as reminders

In addition to icons, many participants use windows as reminders. Card *et. al.* were the first to indicate that windows might be used to remind [5]. Our participants indicated a variety of additional situations not addressed by Card *et. al.* that cause them to use reminder windows but most often these situations were described as interruptions. Indeed, during all of the interviews except one, the participant was interrupted by a phone call or visitor. Whereas in the past lower-power, lower-memory machines could force users to close windows to handle these interruptions, today’s machines allow people to keep many windows open. The power of leaving windows open is that more context of the task that the reminder window represents is readily available to the user. Several participants demonstrated this power when they explained in great detail what different windows reminded them to do.

All three administrative assistants indicated that interruptions are a part of their jobs, whether by face-to-

face communication or through email. All three used email filters to automatically display email messages from “important people” or with “important subjects.” They all had many windows left open to remind them to return to abandoned tasks. Other users also relied heavily on electronic communication, whether email or IM. For example, Participant 5, a multiple monitor user, said “[There are usually] at least six things [in the dock] as reminders to come back to a task [that I have not yet finished]... email is the center of my universe [and] dominates everything I do” (Figure 3). An interesting point is that although information in windows occupying the screen is not used for the task at hand, the display of such information is very important to the user and can aid in remembering to switch tasks. Frequently interrupted users all mentioned a desire to have a visually salient area of the screen to drop windows that should be returned to later, making it less difficult to find abandoned windows. Participant 5 in particular noted that the dock “is not in my face enough sometimes.”

Implication for design: Window managers may consider how to place or alter windows that need later attention. Two immediately obvious ideas are to dedicate a special area for users to drop such windows, or create a “super window” that contains all of the reminder windows and uses change-blind animations to cycle through them. Windows marked as reminders by users could also be graphically altered to gain more prominence in the display.

Implication for evaluation: Increased system power has allowed users to keep many windows open, moving the burden of remembering to complete tasks from the brain to the eyes. Evaluators can assess how well new

techniques allow users to easily have many windows open and how easily reminder windows can be recalled.

Implication for a specific system: This finding and our first finding about windows providing information without having input focus imply that more careful design is needed for systems like CIWM [7], which try to determine the importance of a window based on input focus and close windows deemed unimportant. Closing a reminding window could have a detrimental effect on overall task completion for users.

6.5 The effect of input devices

Throughout our interviews, we found that the type of input devices available to the participants guided the ways that they managed screen space. There are a number of specific examples, but space constraints prohibit discussing all of them. Thus we present the following two typical examples.

Participant 11 uses a laptop system that sits in a docking station on his desk. Attached to the docking station is a flat panel monitor, giving a multiple monitor setup. Initially, he used the touchpad on his laptop. In order to move the mouse between monitors, he needed two to four runs of his finger across the touchpad. Due to this input overhead, the second monitor often contained windows that displayed information but seldom or never received interaction (such as web browsers). Later, he decided to attach a desktop mouse to the docking station, which allowed him to use one motion to switch monitors and prompted him to mix interaction more evenly between the monitors. For example, he now commonly edits documents on the attached monitor while interacting with email about the document on the other monitor.

Consider the case of Participant 12. She has two independent systems on her desk, which means two monitors, two mice, and two keyboards. The desk has a drop-down tray in which one keyboard can be placed. Because of her RSI, she uses the tray for both a keyboard and a mouse. This has two effects: (1) the mouse has very little room to move, which requires her to pick it up and drop it frequently when moving the mouse pointer, and (2) she uses the other system for absolutely nothing but email. For awhile, she ran a program that allows one set of input devices to control multiple systems. However, the amount of pixels that she had to traverse was too large for such a small mouse space: "I tried the one keyboard and mouse, but it didn't work because of the stupid little space for the mouse... I'm limited by the physical desk."

Implication for design: These observations open new avenues for future work. One is to simply study the space management affordances of different devices in order to tailor management techniques, or indeed build devices that more closely match users' characteristics.

Window systems might also track *how* people use the input devices, and provide information on how the user can more effectively use the system or determine display techniques that aid the user's habits.

Implication for evaluation: When evaluating a space management system, researchers should take care to have participants use a variety of input techniques to provide additional reliability of results or determine if some input devices or techniques work poorly with the system. This may take particular importance as advanced input techniques (such as voice commands and eye-gaze tracking) become more robust.

7 Future work

Analysis of our interactions with participants indicates that maximizers may grow extinct as displays support higher resolution at lower cost, and near maximizers could be aided by very simple changes to the maximize operation. We present one design idea that could help careful coordinators with respect to many of the results uncovered in the previous section and some further discussion of multiple monitor systems.

7.1 Rethinking Iconify & Resize

Other than *iconify*, window managers do not have built-in functions that help careful coordinators hide or show specific window contents, which our careful coordinators indicated as quite useful. One idea for diminishing the amount of space occupied by a window is to give the user a simpler way to indicate the parts of a window that should be visible. Consider the IM client window shown in Figure 2. Other than the important list of people who are online, the interface also includes a menu, a logo, an advertisement, a label indicating the buddy list's owner, and two banks of large graphical buttons for interacting with the client, all of which are useless unless the window has input and user focus. Even then, if users want to send messages to buddies, they need only click on the buddies' names to open new IM windows. The *relevant region* of the window is only the buddy list for most of the time that the window is on the screen.

We could diminish the amount of space occupied by the window by simply allowing the user to draw a rectangle about this relevant region, hide the remainder of the window contents, and treat the region as a window unto itself. Note how the standard version of resize would not accomplish the same thing. If a user resizes the window, the only part that becomes smaller is the buddy list, which is exactly the important part. If a user instead just scaled the window to the size of the relevant region, it is unlikely that a user could read the text of the list, thus prohibiting glancing. Finally, the relevant regions concept is an ideal candidate for the window manager because it is user-defined and may

cover sections of UI components, so users could use the regions regardless of applications.

Not only does this concept extend to showing and hiding specific contents (outlined in both subsections 6.1 and 6.2), but it could also help people exploit the use of windows as reminders (subsection 6.4) by showing a small, graphically salient part of the window. Giving users smaller windows to manage can help particularly those who keep some part of the desktop visible (subsection 6.3), since they have a smaller space in which to place their windows.

7.2 Concerning multiple monitors

A common argument given about multiple monitors is that “given enough space, people will tile all windows, so window management becomes less important.” That people specifically hide window contents for a variety of reasons, use “empty space,” and employ space to help remind all cast a shadow of doubt on this belief. Furthermore others have shown that multiple monitors are not treated as simply “additional space” and that people tend to manage space within monitors, rarely allowing windows to straddle physical boundaries [9].

These aspects indicate that techniques that aim to help people manage windows on monitor-sized displays are likely to help users of any number of monitors. Indeed, with more space to manage, techniques such as relevant region-based resizing could dramatically improve the window interaction experience, relieving users of contorting ways of displaying information and accessing windows. Evaluating the ease with which such designs help users both complete multi-window tasks and switch among these tasks will aid in teasing out the specific benefits the designs give. Our findings can aid in developing evaluations that reflect the actual practices and values people exhibit in managing space.

8 Conclusion

While there are many intuitions and beliefs about the way that people manage display space, there is remarkably little documented research on the topic. Our detailed study of actual space management schemes and interaction issues provides such research. We believe that the rise of multiple monitor displays will prompt designers to enhance window management systems. Our analyses will help inform both the design and evaluation of next-generation space management techniques by providing an accurate depiction of the high-level, display-independent techniques that users employ to manage display space.

Acknowledgement

We would like to thank Mary Czerwinski for offering insightful comments about an early draft of this paper.

References

[1] Badros, G.J. *et al.* SCWM: an intelligent constraint-enabled window manager. In *Proc.*

AAAI Symp. on Smart Graphics, 76–83, 2000.

- [2] Beaudouin-Lafon, M. Novel interaction techniques for overlapping windows. In *Proc. UIST*, pages 153–154, 2001.
- [3] Bell, B. and Feiner, S. Dynamic space management for user interfaces. In *Proc. UIST*, 239–248, 2000.
- [4] Bly, S. A. and Rosenberg, J. K. A comparison of tiled and overlapping windows. In *Proc. CHI*, pages 101–106, 1986.
- [5] Card, S. K., Pavel, M., and Farrell, J. E. Window-based computer dialogues. In *Proc. INTERACT*, pages 239–243, 1984.
- [6] Czerwinski, M. *et al.* Toward characterizing the productivity benefits of very large displays. In *Proc. INTERACT*, pages 9–16, 2003.
- [7] Funke, D. J. *et al.* An approach to intelligent automated window management. In *Int. J. of Man-Machine Studies* 38, pages 949–983, 1993.
- [8] Gaylin, K. How are windows used? Some notes on creating empirically-based windowing benchmark task. In *Proc. CHI*, 96–100, 1986.
- [9] Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proc. CHI*, pages 458–465, 2001.
- [10] Henderson, D. A. Jr. and Card, S. K. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. In *ACM Trans. on Graphics* 5(3), pages 211–243, 1986.
- [11] Hutchings, D. R. and Stasko, J. QuickSpace: New operations for the desktop metaphor. *CHI Extended Abstracts*, ACM Press, pages 802–803, 2002.
- [12] Hutchings, D. R. *et al.* Display space usage and window management operation comparisons between single monitor and multiple monitor users. *Proc. Advanced Visual Interfaces*, ACM Press, 2004, (to appear).
- [13] Kandogan, E. and Shneiderman, B. Elastic windows: evaluation of multi-window operations. In *Proc. CHI*, pages 250–257, 1997.
- [14] Macintosh OS X Exposé hot key operations. <http://www.apple.com/macosx/panther/expose.html>
- [15] Myers, B. Window interfaces: a taxonomy of window manager user interfaces. In *IEEE Computer Graphics and Applications* 8(5), pages 65–84, 1988.
- [16] Mynatt, E. D. The writing on the wall. In *Proc. INTERACT*, pages 196 – 204, 1999.
- [17] Ringel, M. When one isn’t enough: an analysis of virtual desktop usage strategies and their implications for design. In *CHI Extended Abstracts*, pages 762–763, 2003.
- [18] Robertson, G. *et al.* The task gallery: a 3D window manager. In *Proc. CHI*, pages 494–501, 1993.