Improving Interfaces for Collaborative Work in Multiple-Device Environments

Jacob T. Biehl and Brian P. Bailey
Department of Computer Science
University of Illinois
Urbana, IL 61801
{jtbiehl | bpbailey}@cs.uiuc.edu

ABSTRACT
Productive collaboration in a Multiple-Device Environment (MDE) requires an effective interface to efficiently manage applications among devices. Though many interfaces exist, there is little empirical understanding of how they affect collaboration. This paper reports results from a user study comparing how well three classes of interfaces; textual, map, and iconic, support realistic, collaborative activities in an MDE. From empirical results, our observations, and an analysis of how groups interacted with each interface, we produced a set of design lessons for improving management interfaces for MDEs. The lessons were demonstrated within the iconic interface, but are just as applicable to other interfaces. This work contributes further understanding of how to design effective management interfaces for MDEs.

Author Keywords
Collaboration, Iconic interface, Interactive workspace.

ACM Classification Keywords
H.5.2. User Interfaces, H.5.3 Group and Organizational Interfaces, I.3.6 Methodology and Techniques.

INTRODUCTION
To work productively in a Multiple-Device Environment (MDE), users need effective user interfaces for quickly and easily managing applications among devices [9, 14, 23]. To support effective group brainstorming [35], for example, users need to rapidly sketch alternative ideas and spread those ideas across devices. If the interface cannot support, or otherwise disrupts rapid exchange of ideas, it will hinder rather than facilitate individual and collaborative work.

Within an MDE, the term *device* is used to refer to a laptop, tablet, large display, etc., each driven by an independent, but networked system. Our work empirically compares how three classes of interfaces; *textual*, *map*, and *iconic*, support management of applications for problem solving activities in an MDE. These three classes of interface were selected since they have been commonly used in existing MDEs.

A textual interface provides identifiers for applications and devices in an MDE (e.g., [23, 24]). To relocate applications, users select the applications and the source and destination device, e.g., by selecting IP addresses from text-based UI controls. This interface offers simple interaction, but users must recall how identifiers map to applications and devices.

In a map interface (e.g., [21]), users are provided with a strict top-down view of the workspace. Users are able to use their spatial reasoning abilities to identify devices based on their location in the environment. However, users must still map textual identifiers to corresponding applications.

Our work has been investigating the use of iconic interfaces for managing applications in MDEs [5, 7]. An iconic interface provides a world-in-miniature representation of a workspace in a 2-D, fold-out view. In this representation, devices match their spatial configuration in the workspace and representations of the applications are shown within each device. As with the map interface, users are able to use their spatial reasoning abilities, but interact with visual as opposed to textual representations of applications [17].

This work empirically compares how well these 3 classes of interfaces support realistic, collaborative tasks in MDEs. We configured a representative MDE consisting of three tablets and two large displays and asked groups of users to perform creative problem solving activities within it. Users needed to plan the activities, coordinate actions, exchange artifacts (applications), and transition between individual and shared work. Each group performed a similar activity with each of the management interfaces. We measured time to relocate each application, workload, and satisfaction, and observed how groups used the MDE to structure their tasks.

Results showed that users were able to perform relocation tasks faster with the iconic interface and it was preferred over the others. However, iconic interfaces may not always be the most effective interface for an MDE. For example, if the location of devices changes often due to users moving around in the workspace with their devices, it would be cumbersome to keep the spatial representation up to date.

Thus, from the empirical results, observations of how users structured their activities in the MDE, and analysis of how users interacted with each interface, we produced a set of design lessons for improving management interfaces. These lessons were practically demonstrated within the iconic interface, but analogous improvements could be made to the others. Our work contributes further understanding of how to design effective management interfaces for MDEs.

RELATED WORK

We describe potential benefits of MDEs, relate interaction techniques for MDEs to those in virtual window managers, review current mechanisms for managing applications in MDEs and evaluations of these mechanisms, and situate our work in the broader area of groupware for co-located users.

Multiple-Display Environments

A multiple-display environment (MDE) refers to a physical workspace in which numerous shared and personal devices are networked together. The vision is to enable individual and especially small groups of users to seamlessly create, share, and juxtapose digital artifacts across devices during creative problem solving activities [14, 23, 25].

To provide the necessary software infrastructure, several distributed OSes such as Gaia [23], iROS [13] and Aura [27] have been developed. The systems allow independent, networked devices to form a single, virtual system. Our works seeks to develop effective interfaces and interaction techniques that allow users to quickly and easily manage applications and input across devices in an MDE.

Virtual Desktop Window Managers

Virtual desktop managers such as ROOMS [12], FVWM [1], KDE [2], and Sawfish [3] enable applications to be managed across multiple virtual desktops. Interfaces for MDEs extend the concept of a window manager to multiple physical screens separated in a physical workspace.

Mechanisms for Managing Information in MDEs

Many user interfaces and interaction techniques have been developed for managing applications among independent devices. The I-Land project [31] produced several interactions such as shuffle, throw, take, and pick-and-drop for relocating applications within large screens and between screens. In Easy Living [9], the managing infrastructure actively tracks users in the workspace and automatically relocates applications to devices closest to the user.

With UbiTable [25], ConnecTables [33], and augmented surfaces [22], users are able to share applications among personal devices as well as shared workspaces. The main interaction technique for sharing applications is the use of a virtual path, where users drag an application to the edge of a screen causing it to appear on the screen of an adjacent device.

PointRight uses unified geometric paths to enable input redirection across devices [13]. Users can move the cursor implicitly (without a UI control) among local and shared devices, and interact with applications. MightyMouse offers a similar concept of input redirection, but provides a control panel with buttons positioned to approximate the spatial layout of the corresponding devices in the workspace [8].

In iCrafter [16], a user relocates an application by migrating the service that supports it to another device. Using an interface that provides a strict top-down view of the workspace, the user drags a textual identifier of the service and drops it onto the destination screen. The map interface in our study was designed to typify this interaction design.

In [16], researchers extended a Web browser to enable users to relocate browser windows across machines. Part of the interaction involved selecting a textual identifier of the destination screen from a list of available choices. The text interface used in our study represents this interaction.

Our work has been investigating the use of iconic interfaces for managing applications and input in MDEs [4, 5]. The interface representation provides a 2-D, foldout view of the workspace where the walls appear pulled down, allowing all applications to be immediately visible. Our most recent implementation, SEAPort [6], uses this same basic spatial representation, but provides a fan-out view for selecting occluded representations, zooming for interacting with small representations, and portal views to determine which artifacts are on which screens, important for facilitating workspace awareness [10]. SEAPort was used in our study to represent the broader class of iconic interfaces for MDEs.

Our work compares three specific, but representative user interfaces (text, map, and iconic) for managing applications in MDEs for collaborative activities. To the best of our knowledge, this is the first comparison of interfaces for collaborative tasks situated within an MDE. However, the design lessons distilled from the study can be applied to the broader class of management interfaces for MDEs.

Evaluations of Existing Mechanisms

There have been many usability evaluations of interfaces for MDEs (e.g., [15, 16, 26]). The studies have focused on evaluating the effectiveness of a specific interface, whereas our work focuses on comparing *alternative* interfaces.

Naceta et al. compared several reaching techniques within an MDE [20]. In their study, individual users performed a rapid sequence of prescribed relocation tasks. Another study conducted as part of the first author's M.S. thesis [4] tested how well individual users could perform relocation and redirection tasks with several management interfaces.

Our work substantially extends this direction of research by comparing how well groups of users can utilize alternative management interfaces for performing *collaborative* and *loosely-structured* activities within an MDE. From our results, we produced a set of design lessons that are applicable to the broader class of management interfaces for MDEs.

Groupware for Co-Located Users

Co-located groupware has included the use of both single and multiple displays. Single display groupware allows co-present users to concurrently interact with applications on a single shared workspace. For example, KidPad allows two or more children to draw on the same canvas [29]. Pebbles allows multiple users to interact with the same application on a remote display through local PDAs [19]. In contrast, MDEs allow users to retain affordances of personal devices for individual work and large displays for shared work.

Systems such as CoLab [28] allow instances of shared applications to run on personal devices, allowing seamless transitions between individual and shared work. Similarly, Xia et al. showed how MS Office applications could be extended to support collaborative editing [37]. MDEs allow users to collaborate with applications by managing them across devices. As support for relocating applications will soon be common (e.g., with .NET and similar frameworks), our focus is on developing effective user interfaces and interaction techniques for managing applications in MDEs.

COMPARATIVE STUDY

Our study was designed to answer these questions:

- How does the management interface affect relocation performance, subjective workload, and user satisfaction when performing collaborative tasks in MDEs?
- How do users structure their activities in MDEs and how does the management interface affect that process?
- What are the strengths and weaknesses of alternative interfaces for supporting collaboration in MDEs and what lessons can we learn that would improve these interfaces?

Experimental Design

The experiment used a mixed design with Interface (Text, Map, and Iconic) as a within-subjects factor and Activity (Comic strip and Collage) as a between-subjects factor.

Users

18 users participated in the study in groups of three. Users consisted of undergraduate and graduate students, and administrative professionals from our institution. Ages ranged from 18 to over 40. Users were compensated with a $5 gift certificate to a local coffee shop for participating. None of the users had prior experience with MDEs.

Multi-Device Environment

As shown in Figure 1, our MDE consisted of two 61” plasma screens each driven by an independent machine and three Tablet PCs. The tablets were positioned 2’ apart along one side of a meeting table and had resolution 1024x768. Each user was seated in front of a tablet and used a stylus for input. The two plasma screens were positioned behind the table, 1’ apart on the same plane, and were within the users’ field of view. Their resolution was 1360x768.

Figure 1. A group of users performing the collage activity in our MDE. Each user is individually searching for images using several open Internet Explorer windows on a tablet PC. When a desirable image is found, the user relocates the shared canvas (shown on the right-most large display) to the local tablet to add the image. The user may then relocate the canvas back to a large display or directly to another user’s tablet.

An MDE can be composed of any number of shared and personal devices, where the devices can have screens of widely different sizes and the devices can be arranged in myriad physical configurations. The MDE used in our study was specifically configured to be representative of the broader class of MDEs designed to support small work groups (about 2-6 users), as exemplified in [14, 23].

In our configuration, only two (as opposed to three) large displays were used to prevent the MDE from having the same number of personal and large displays, which we felt would generate more interesting patterns of use. Having 3 users in each group is representative of small work groups.

Distributed Drawing Canvas

We created a customized distributed drawing canvas to be used for both of the experimental activities (Figure 2). The application supports basic inking and editing commands using stylus or mouse input. Images can be placed on the canvas through a drag and drop interaction. A lightweight

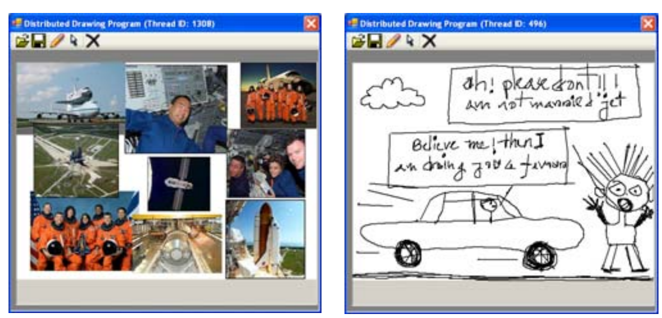


Figure 2. A sample of task artifacts created during the study. The left shows a collage created during a collage activity while the right shows one frame from a comic strip activity.

service executes on each device in the MDE, allowing the canvas to be relocated. When relocation is requested, the service on the source device serializes the canvas and sends it over a network connection to the target. The service on the target device then unpacks the content and recreates the canvas. Over a 100Mbps Ethernet, relocation takes less than 250ms. Many commercial applications may soon have the ability to be seamlessly migrated across machines, as this support is being integrated within many common runtime architectures (e.g., .NET).

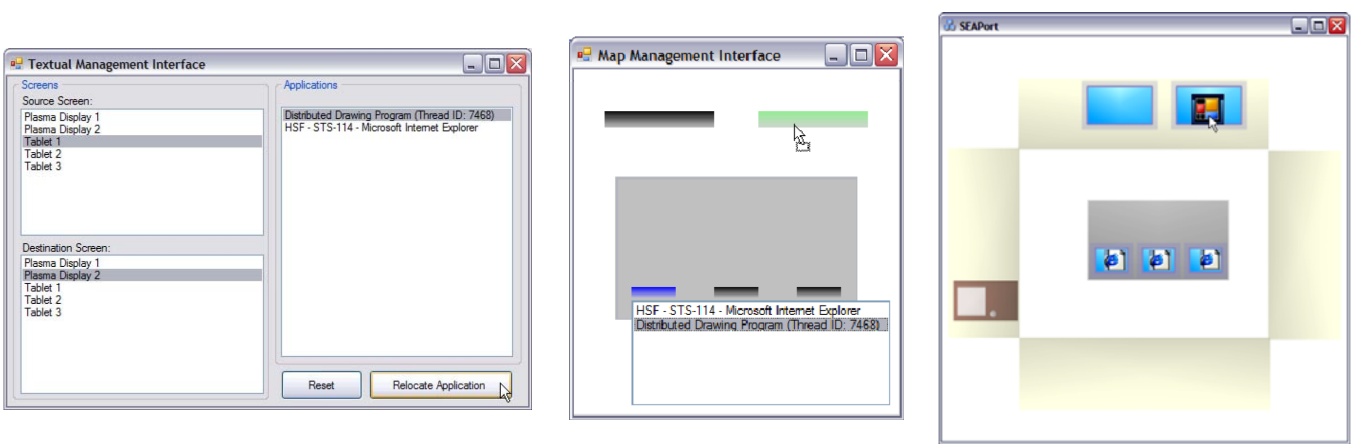
Management Interfaces

Three management interfaces were compared in the study:

- *Textual*. As shown in Figure 3a, the textual interface was composed of selection lists and command buttons. To relocate an application, a user selects the source screen, application to relocate, the destination screen, and then selects "Relocate Application." Selection of the application and devices is made from a list of textual identifiers. For the applications, the identifiers matched the text used in their title bar. For the devices, identifiers matched physical labels that were attached to the devices, e.g. "Plasma Display 1", which were legible from a distance.
- *Map*. As shown in Figure 3b, the map interface provides a strict top-down view of the MDE. Each device is represented as a thin rectangle with the tablets being shown on the table. The placement of the representations matches the position of the devices within the physical workspace. To relocate an application, a user selects the representation of the source device and a drop-down list appears showing a textual list of running applications. Identifiers in the list match the text in the applications' title bars. A user selects and drags the desired text item to the representation of the target device, which flashes locally to indicate the action has been completed.
- *Iconic*. As shown in Figure 3c, the iconic interface provides a 2-D, foldout view of the environment [6]. The walls appear pulled down and each device faces upwards. Each application is represented with its icon sized and positioned relative to its size and position on the physical device. To relocate an application, a user selects an icon, drags it to the desired device, and positions it. If the desired icon is occluded, the user can right-click on the screen and the icons fan out into a non-occluded view, from which the desired icon can be selected. A user can also zoom into a particular device by performing a pullout gesture on one of its corners. Once zoomed in, the icons change to thumbnails with live updates. If a user drags a thumbnail beyond the edge of the device's representation, both the application and screen quickly animate back to their original size and the relocation interaction can be completed without a mode switch.

When a specific interface was assigned, an instance of that interface was executed on each user's tablet in the lower right corner. This was done to limit the interaction overhead of having to access the interface from a menu or other control. Our experience suggests that this is similar to how the interfaces would be configured and used in practice. Each interface went through at least one round of usability testing prior to the study to ensure a fair comparison.

The interfaces chosen typify interaction designs used in many existing MDEs or similar environments. The textual interface typifies interaction designs in MB2Go [16], iROS [13], and Gaia [23], where at least part of the interaction is to select textual identifiers of applications or devices. The map interface typifies interaction designs in iCrafter [14] and Mighty Mouse [8], where the visual representation in the interface reflects the spatial arrangement of the physical devices. The iconic interface typifies interaction designs in 2-D world-in-miniature interfaces, e.g., [5, 6]. These interfaces were first popularized for navigating 3-D worlds [30], but offer a promising interaction metaphor for MDEs.



(a) Textual Interface. (b) Map Interface. (c) Iconic Interface.

Figure 3: The three interfaces used in our study. The interfaces are shown from the perspective a user sitting at the table. Each is currently showing a user in the process of relocating an application from the tablet on the left to the large display on the right.

Our study did *not* include the use of a virtual path interface, the extension of a multi-monitor interaction, as it is often not a practical solution for an MDE. Though used in prior work, e.g., [13, 22, 25, 33], this technique requires the cursor to be controlled beyond the user's local screen, necessitating the use of a *relative* input device. Thus, this technique does not support the use of *absolute* devices such as stylus and touch input, which are prevalent in MDEs. Each interface tested in our study can support mouse, stylus, and touch input.

Collaborative Activities

Two collaborative activities were designed for the study:

- *Create a digital collage*. The purpose of this activity was to produce a meaningful digital collage about a recent news event, e.g., NASA's return to space. Each user's tablet had Internet Explorer running which showed a unique set of links to relevant news sites as its homepage. Users visited the sites searching for images related to the given theme. When an appropriate image was found, the user found and then relocated the shared collage (an instance of the distributed drawing program) to his tablet, dragged the image to the collage, sized and positioned it, and then optionally relocated it to another device. The final collage was to contain at least nine images and each user was asked to contribute at least three.
- *Create comic strips*. The purpose of this activity was to sketch two separate comic strips, each with three frames. Each frame corresponded to an instance of the drawing canvas. To seed the group's creative thinking, for each comic strip, we provided content for one frame and asked the group to sketch the other two and add dialog to all three. For example, one given frame had a rough sketch of a person crossing the street. At the start of the activity, six drawing canvases were executing on the two large screens and each canvas was to become one frame of a comic. To facilitate the need for coordination, each user was asked to choose 1 of 3 responsibilities; creating characters, creating scene content, and adding dialogue. However, if desired, the group could devise an alternative work plan.

These activities are representative of creative problem solving tasks that small teams of users often engage in and are the types of activities MDEs are designed to support. Users had to plan the activity, coordinate actions, exchange digital artifacts, and transition between individual and group work. The tasks also required different degrees of coordination. For example, for collage generation, users needed to coordinate access to just one shared resource whereas creating comics required coordinated access to multiple resources based on functional roles. By engaging users in these activities, we were able to study how users leveraged personal and shared devices to exchange artifacts, coordinate actions, maintain awareness, etc. and how alternative management interfaces affected those behaviors.

Procedure

When users arrived at the lab, we went through an informed consent process. Users were then trained on the drawing tool, given general instructions for the assigned activity, and introduced to the first interface. Three groups (each with three users) performed the collage activity and three groups performed the comic strip activity. Each group performed the assigned activity once with each interface. Interfaces were presented using a Latin Square design.

The group was given 15-20 minutes to perform the activity with an interface and then completed a subjective workload and interface questionnaire. This process was repeated two more times for the other interfaces. Subsequent activities were varied slightly, e.g., a different theme and list of web sites was provided for collage generation and different seed sketches were provided for the comic strip activity.

After using the third interface, the experimenter led a group discussion about using all 3 interfaces. Camtasia was used to record users' screen interaction and the entire session was video taped. The session lasted about 75 minutes.

Measurements

For both activities, we measured:

- *Time to relocate an application*. This was measured from when a user made a directed action with the interface to when the application appeared on the destination screen. Measurements were computed from time stamps in the screen interaction videos.
- *Subjective workload*. This was measured using the NASA TLX [11], which measures workload along 6 dimensions: *mental demand*, *physical demand*, *temporal demand*, *own performance*, *effort*, and *frustration*. Responses are marked by drawing a vertical line along a continuous scale from Low to High, measured in 1/16" increments.
- *Satisfaction*. Users rated each interface according to simplicity, comfort, awareness, and satisfaction. Ratings were structured using a 7-point Likert scale (7 was most positive) where statements were made in neutral form.

These measures are representative of those used in prior work [4, 20]. In addition to these quantitative measures, we also wanted to observe how users interacted with the MDE, the interfaces, and each other during the collaborative tasks.

RESULTS

We discuss qualitative results derived from observing how users structured their activities in the MDE, open-ended responses from questionnaires, and group discussion. Then, quantitative results for performance, subjective workload, and user satisfaction are compared among the interfaces.

Qualitative Results

In this section, we first discuss how the groups structured their activities, detailing how work was divided among users, how personal and shared devices were utilized, and how users coordinated individual efforts. We then discuss salient qualitative differences among the interfaces.

All groups were able to successfully complete the activities using each interface in the allotted time. This shows that groups can meaningfully collaborate in MDEs. Several users noted that these types of environments seemed well suited for performing collaborative tasks, e.g., one user stated "I can definitely see this workspace being used for this type of task." Another added that she would want to work on her group-based class projects in this environment.

When starting an activity, groups would devise an overall plan and discuss how to best divide the work. For example, when creating comics, users would develop a shared vision for the strips and decide which person would draw which parts. For the collage, planning entailed determining an initial order for passing the collage (though it was rarely followed). Periods of individual work and group discussion were seamlessly interwoven during the activities, showing that the use of MDEs can effectively support this common and important component of collaborative work [34].

Groups leveraged the large displays to provide a shared workspace to discuss intermediate outcomes and organize workflow. For example, once a user finished their part of a comic's frame, they would relocate it to a large display, position it on the screen so that it was closest to the person they thought needed it next (if possible with the interface), and tersely announced its availability. At the end of the activity, groups typically used the large displays to organize the final artifacts, e.g., separating frames for each comic between the displays. Likewise, when creating the collage, users would sometimes relocate it to a shared display to show the group an interesting image or to discuss if the selected image was an appropriate fit for the given theme.

Users would also exchange task artifacts directly between their personal devices. For example, during the collage activity, a user would add an image to the collage and then keep it locally. When another user wanted it, they would call out whether they could have it. If so, the current owner would relocate the collage to the requesting user's device (push). In a few cases, a user would relocate the collage from another user's device to his local device (pull), but this was generally considered socially inappropriate.

One difference among the interfaces was the amount of workspace awareness (which artifacts are on which screens) that was immediately visible, which affected group behavior. When using interfaces with less awareness information (e.g., the map and textual interfaces), users compensated by verbalizing more awareness updates (e.g., "I am taking frame 3 now") and requests (e.g., "who has the collage?"). Most users found these verbal broadcasts to be very disruptive to their individual work. This finding indicates that it is necessary to visualize an adequate level of workspace awareness in a management interface, despite the fact that users are co-located. This was one reason why users preferred the iconic interface over the others, as they could extract awareness information with a quick glance.

Another qualitative difference was in how the interfaces affected a user's perception of the workspace. For example, users commented that with the iconic interface, the workspace seemed much more cohesive than when using the other interfaces and that the iconic interface made it much more "inviting" to use the shared displays. This is important since a driving motivation for using MDEs is to facilitate sharing of information among group members.

Quantitative Results

A MANOVA showed that Interface did not affect ratings of subjective workload, though the trends favored the iconic interface (36.9%, 46.2%, and 42.8% of maximum workload for the iconic, map, and textual interfaces, respectively). This result shows that users did not find any one interface to be substantially more demanding to use than the others.

An ANOVA showed that the interface had a main effect on how quickly users could relocate applications ($F(2,324)=51.64$, $p<0.001$). Post hoc analysis showed that a user relocated applications faster with the iconic interface ($\mu=2.52s$, $sd=0.63$) than the map ($\mu=5.78s$, $sd=0.47$; $p<0.001$) and textual interface ($\mu=12.12s$, $sd=0.71$; $p<0.001$). The iconic interface thus provides a meaningful performance improvement over the map and textual interfaces (~56% and ~79%, respectively). Users could also perform relocations faster with the map interface than the textual interface ($p<0.001$), which is also a meaningful performance improvement (~52%).

The difference between relocation times for each interface is only a few seconds. This may be of little concern if relocations are performed infrequently during a collaborative activity. But, when a group is deeply engaged in creative problem solving, they would want and need to frequently exchange artifacts associated with alternative ideas among devices as quickly as possible (e.g., when debating alternative outlines for the results section of a paper). In these cases, the cumulative effect of these small differences could severely inhibit the free and rapid exchange of alternative ideas, which is crucial during the creative process [28].

An action analysis [18] showed that the differences in performance cannot be explained simply by differences in the number of operators, as each interface required about the same number of steps (4-5). We attribute the differences to "think" time, where users are bridging the semantic gap between the state of the workspace (which applications are where) and the representation in the management interface. For example, when performing relocation tasks in the map and textual interfaces, analysis of the videos showed that users would pause for a few seconds to determine which applications/devices mapped to which identifiers, while their interaction was more fluid with the iconic interface.

An ANOVA showed that Interface affected ratings of simplicity ($F(2,34)=4.46$, $p<0.019$), comfort ($F(2,34)=4.24$, $p<0.023$), awareness ($F(2,34)=6.42$, $p<0.004$), and satisfaction ($F(2,34)=4.65$, $p<0.016$). Post hoc analysis showed that users rated the iconic interface higher along each dimension (μ=5.89, 6.00, 5.44, 5.18, respectively) than the map interface (μ=4.94, 5.06, 4.28, 3.78; $p<0.02$, $p<0.03$, $p<0.05$, $p<0.03$, respectively). Users were also more satisfied with the iconic interface (u=5.18) than the textual interface (μ=4.17, $p<0.05$). No other differences were detected. Results indicate that users had a reasonably strong preference for the iconic interface over the other interfaces.

DISCUSSION

Results showed that, compared to the other interfaces, the iconic interface enabled faster relocation of applications and better awareness of the workspace without inducing a measurable increase in workload. The iconic interface was also the interface most preferred by users. The differences between interfaces are likely due to the iconic interface providing a more comprehensive spatial and visual representation of the workspace than the others. Visual and spatial information is typically processed much faster than text, as text requires more downstream processing [36].

To be effective, an iconic interface must obtain information about the spatial arrangement of devices in the workspace. This becomes especially difficult when devices participate only briefly in the MDE or participate after the initial workspace representation has been defined. One solution is to leverage our existing configuration tool which integrates with the interface runtime to allow the spatial representation to be dynamically constructed and modified on the fly [7]. For example, just before the start of a collaborative activity, the tool could be used to configure the spatial layout of participating devices within the interface. Another solution, similar to [25], is for a device to connect to an established MDE session. Once connected, a representation of the device would appear in each user's iconic interface which could then be manually positioned as desired.

Though we believe that iconic interfaces would be effective for many MDE configurations and group activities, it may not always be the most effective interface. For example, if the locations of devices are constantly changing due to users moving around in the workspace with the devices, it would be cumbersome to keep the spatial representation up to date. Likewise, if users wanted to manage applications on a device with limited display size, such as a PDA, the iconic interface would be less effective.

Thus, in the next sections, we discuss design lessons that can be generally applied to the class of MDE interfaces. These lessons were derived from observations of how users structured their activities in the MDE as well as how they interacted with all three interfaces. We also describe how the lessons can be implemented within the iconic interface.

Design Lessons for Management Interfaces

From the study, we learned the following lessons about how to better design management interfaces for MDEs:

L1. Provide a view that allows all applications to be seen at once. When using the textual and map interfaces, users often explored each device's content in search of a specific application. Several users commented that they wanted to be able to glance at any device and know what was running on it, as they could with the iconic interface. For example, one user stated "the iconic interface was nice because it allowed you to see everything at once." Other interfaces could offer analogous design features. For example, the map interface could have an interaction which toggles the drop-down lists of applications for all the devices at once, allowing a more holistic view of the workspace.

L2. Allow users to spatially position the application on the destination screen. The textual and map interfaces placed a relocated application in the middle of the screen on the destination device. Almost every user expressed a desire to control where on that screen the application would appear, pointing out that this was one of the best features of the iconic interface. The textual and map interfaces could implement an analogous interaction, e.g., the interfaces could enable a quadrant of the destination screen to be selected or could show a small representation of a device's screen and allow the user to position an outline within it. Such an interaction would address the lesson, while still maintaining the basic metaphor of the interface.

L3. Allow an application running on a personal device to show a mirrored copy (shadow) on a shared display. The granularity of coordinating at the application level in MDEs can be too large. Users often wanted moment-by-moment awareness of what other users were doing so they could coordinate their own actions and creative thinking. For example, during the comic strip activity, a user would periodically become "stuck" because they did not know what else to draw until they saw the content being created by another user. This often required the user to wait for related artifacts to be relocated to a shared display. This delay was often too long. Users expressed a desire to create a shadow of an application to be shown on one of the shared displays. The shadow would allow a user's moment-by-moment interaction with the application on his personal device to be visible to the group. But, the application could only be controlled by the owner of the local device.

L4. Allow users to assign ownership or role-based identifiers to representations of devices. At the start of an activity, groups would devise a work plan and agree upon how screens would be used and the roles that users would fulfill. Since this information had to be retained in short term memory, it was often forgotten and had to be periodically reacquired by asking the group. User comments reflected the inability to relate devices to the context of the task, as one user stated "identifiers were missing that personal identification." One solution would be to allow groups to externalize part of the task context into the interface itself. For example, in the map or iconic interface, users could configure a label for each screen, such as "Comic 1" for the left shared display or "Character design" for the personal device (i.e., user) assigned to fulfill that role.

L5. *Provide visual feedback in the interface of ongoing interactions of other users*. Especially during the collage activity, we often observed two or more users attempting to relocate the same application at the same time, but to different destinations. In these situations, the user who completed the interaction first had his relocation performed, while the others were left confused about why the application did not move to the device they selected. Users shouted "where did it go" or "who took the frame." This shows that although users are co-located, MDE interfaces need to provide feedback of other users' ongoing interactions. For example, when a user selects and drags an application's representation, other users' interfaces could highlight the ongoing interaction with a user-identifying color.

L6. *Allow control over whether applications will appear in other users' interfaces.* Though each application was part of the shared activity in our study, several users raised concerns about privacy when using the MDE for other activities. Specifically, they were concerned that there would be situations when they would not want other users to be able to see which applications they were running. Even the limited information provided in the interfaces, such as an application name or icon, may divulge too much information (e.g., "Outlook" easily gives away that a user is reading email). Users should be able to control whether applications appear in a management interface and how much detail is shown (e.g., which of an application's name, icon, and thumbnail can be shown).

L7. *Allow users to **place** applications onto, but not **take** applications from other users' personal devices*. During the activities, users would occasionally relocate (*take*) applications from another user's personal device to their own. Users emphatically disliked having an application taken from them, even if they were not currently using it, unless permission was given. Users stated that having an application taken was "annoying" and an "invasion" of their personal space. However, users stated that having an application *placed* onto their personal device was perfectly acceptable, as long it appeared behind the focus application (next lesson). To prevent taking, an interface could allow a user to "pin" an application to the local screen, which would not allow other users to relocate it using their interface. Users further stated that "pinned" should be the default setting for applications running on personal devices.

L8. *Do not position the application that was relocated in front of the focus application on the destination screen*. In each interface, when an application was relocated, it was placed in front of the existing applications. This seemed reasonable when designing the interfaces, but users were frustrated when an application suddenly appeared and disrupted their ongoing individual or shared work. Consistent with *L2*, part of the solution is to allow users to position the application off to the side of the destination screen such that it does not visually interfere with the focus application. However, this is not always possible due to limitations of screen space and application size. Thus, the solution should also include setting the z-order of the in-transit application such that it appears *behind* the focus application.

L9. *Provide enough awareness in the interface such that users do not need to compensate with verbal protocols*. When using the textual or map interfaces, users often searched devices looking for a specific application or would ask group members where it was. The latter interrupted ongoing work, which many users found to be annoying. These types of inquiries occurred much less often with the iconic interface. This illustrates that communicating workspace awareness (which artifacts are on which devices [10]) in interfaces for MDEs is necessary, despite the fact that users are co-located.

Improving Management Interfaces for MDEs

We next demonstrate how many of our lessons were used to improve the iconic interface used in the study. We chose the iconic interface for improvement since it was shown to be effective and is our specific area of interest. However, the lessons are just as applicable to the other interfaces.

We addressed L4-8 by adapting our iconic interface – SEAPort. This interface already supports the first two design lessons while the third remains work in progress, as it requires building support into the underlying application and systems software. This support could be provided by integrating functionality similar to WinCuts [32]. L9 is implicitly addressed through the others. The interface solutions described next are just a subset of those possible.

(L4) To integrate this lesson, we modified SEAPort to allow users to configure labels or add user icons to the devices (see Figure 4). The user icons are similar to buddy icons used in popular instant messenger applications. The icons allow users to personalize how their device appears in the interface. Users can also add labels to devices to express the specific role of that device relative to the task or use a label for a personal device in place of a user icon. Because the labels do not alter or obscure other items within the interface, their addition does not diminish the affordances of the visual representation provided in the iconic interface.

(L5) To provide feedback of ongoing interactions, the interface now mirrors relocation actions in each instance. For example, suppose an application is being relocated by a

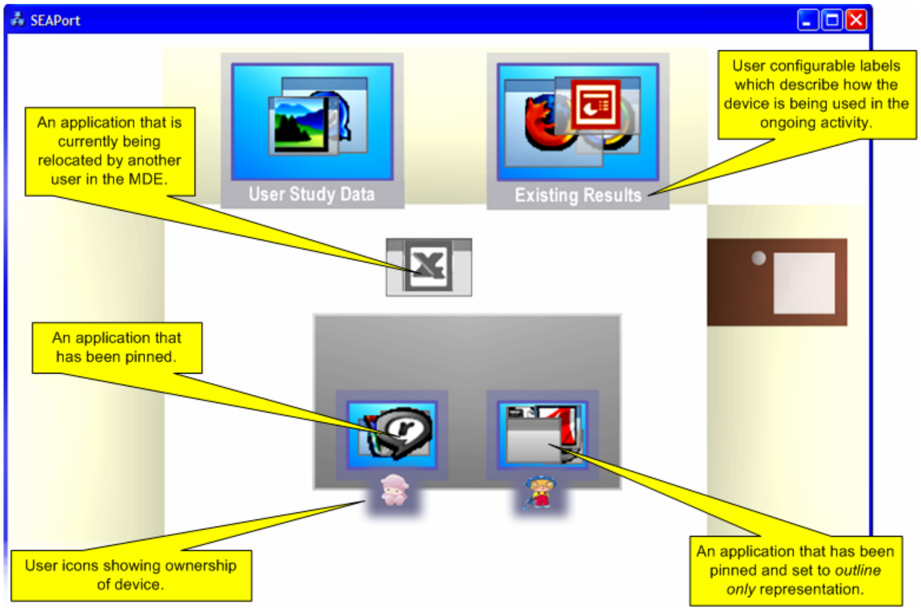


Figure 4. The iconic interface with callouts explaining the improvements derived from our design lessons.

user. As shown in Figure 4, to indicate that the application (an instance of Microsoft Excel in the figure) is being relocated by another user, the application's representation in the local interface appears grayed out and is shown moving between devices. By showing actions *as* they occur, we not only provide better awareness of the workspace, but also prevent users from performing conflicting actions.

(L6-L7) We hooked into Windows to add two additional buttons to a window's title bar (Figure 5). The pushpin allows users to toggle whether other users can relocate the application from their personal device. In the pinned position, the application cannot be relocated, but can be relocated in the un-pinned position. The local user is always able to relocate the application. When an application is placed on a shared device, the pushpin disappears.

The second button allows users to control the privacy of local applications. When a user clicks this button, a drop down menu appears. The menu presents four options for defining how the application is represented in other users'

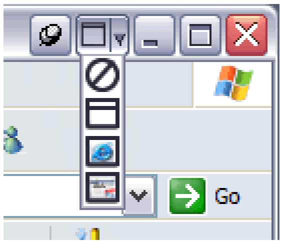


Figure 5. Two additional buttons appear on the title bar of each application. The leftmost button allows users to "pin" an application so that it cannot be relocated by other users. The adjacent button allows the user to specify how the application can be represented in other users' interfaces. The options are *invisible, outline only, icon*, or *thumbnail* views.

interfaces. The options are; *invisible* (no representation is shown), *outline* (a rectangular outline is shown), *icon* (the application's icon is shown), and *thumbnail* (icons are shown and live thumbnail views are permissible).

(L8) This lesson was addressed by changing the z-order of the application to the top level minus 1 as soon as it appears on the destination screen. Thus, if the in-transit application is placed such that it overlaps with the application in focus, it will appear just behind the one in focus.

These lessons facilitated significant design improvements to the iconic interface – improvements that would probably not have been considered otherwise. Though we illustrated how the lessons could be applied for only the iconic interface, they are just as applicable to the other interfaces studied as well as to the broader set of interfaces for MDEs.

CONCLUSION AND FUTURE WORK

A basic challenge in realizing the potential of multi-device environments (MDEs) is developing interfaces that allow applications to be managed among devices. In this paper, we reported empirical results from a user study comparing three alternative interfaces for managing applications in an MDE. Results showed that users preferred and performed better with the iconic interface, due to its comprehensive spatial representation. From an analysis of how users interacted with all three interfaces, we distilled new lessons about how to design interfaces for MDEs and demonstrated the implementation of these lessons in an iconic interface.

We have three directions for future work. First, we want to more fully implement the lessons in our iconic interface and evaluate its improved efficacy. Second, we want to evaluate how using an MDE compares to using other collaborative environments such as a wall-size display or single display groupware system for group-oriented tasks. Finally, we want to conduct field studies to understand the strengths and weaknesses of MDEs and management interfaces when used for practical work over a longer duration of time.

REFERENCES

1. Fvwm.
2. Kde.
3. Sawfish Window Manager.
4. Biehl, J.T. Comparing a Textual Interface, Virtual Paths Interface, and Iconic Map Interface for Effective Interaction in an Interactive Workspace. Department of Computer Science, University of Illinois at Urbana-Champaign, 2004.
5. Biehl, J.T. and B.P. Bailey. ARIS: An Interface for Application Relocation in an Interactive Space. *Graphics Interface*, 2004, 107-116.
6. Biehl, J.T. and B.P. Bailey. Improving Scalability and Awareness in Iconic Interfaces for Multi-Device Environments. *In submission, CHI*, 2006.
7. Biehl, J.T. and B.P. Bailey. A Toolset for Constructing and Supporting Iconic Interfaces for Interactive Workspaces. *INTERACT*, 2005, 699-712.
8. Booth, K.S., B.D. Fisher, C.J.R. Lin and R. Argue. The "Mighty Mouse" Multi-Screen Collaboration Tool. *UIST*, 2002, 209-212.
9. Brumitt, B., B. Meyers, J. Krumm, A. Kern and S.A. Shafer. Easyliving: Technologies for Intelligent Environments. *Handheld and Ubiquitous Computing*, 2000, 12-29.
10. Gutwin, C. and S. Greenberg A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Journal of Computer-Supported Cooperative Work* (3-4), 411-446.
11. Hart, S.G. and L.E. Stateland. Development of NASA-TLX (Task Load Index): Results of Emperical and Theoretical Research. Hancock, P.A. and Meshkati, N. eds. *Human Mental Workload*, North-Holland, Amsterdam, 1988, 139-183.
12. Henderson, A. and S.K. Card 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.
13. Johanson, B. and A. Fox. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. *WMCSA*, 2002.
14. Johanson, B., A. Fox and T. Winograd. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms *IEEE Pervasive Computing*, *1 (2), 67-74*.
15. Johanson, B., G. Hutchins, T. Winograd and M. Stone. Pointright: Experience with Flexible Input Redirection in Interactive Workspaces. *UIST*, 2002, 227-234.
16. Johanson, B., S. Ponnekanti, C. Sengupta and A. Fox. Multibrowsing: Moving Web Content across Multiple Displays. *UbiComp*, 2001, 346-353.
17. Johnson, J., T. L. Roberts, W. Verplank, D. C. Smith, C. H. Irby, M. Beard, and K. Mackey The Xerox Star: A Retrospective. *IEEE Computer*, *22* (9), 11-26.
18. Lewis, C. and J. Rieman. Task-Centered User Interface Design, 1994.
19. Myers, B.A., H. Stiel and R. Gargiulo. Collaboration Using Multiple Pdas Connected to a Pc. *CSCW*, 1998, 285-294.
20. Nacenta, M.A., D. Aliakseyeu, S. Subramanian and C. Gutwin. A Comparison of Techniques for Multi-Display Reaching. *CHI*, 2005, 371-380.
21. Ponnekanti, S.R., B. Lee, A. Fox, P. Hanrahan and T. Winograd. iCrafter: A Service Framework for Ubiquitous Computing Environments. *UbiComp*, 2001.
22. Rekimoto, J. and M. Saitoh. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. *CHI*, 1999, 378-385.
23. Román, M., C. Hess, R. Cerqueira, A. Ranganat, R. Campbell and K. Nahrstedt. Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing, 1 (4), 74-83*.
24. Schilit, B.N., N.I. Adams and R. Want. Context-Aware Computing Applications. *WMCSA*, 1994, 85-90.
25. Shen, C., K.M. Everitt and K. Ryall. Ubitable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. *UbiComp*, 2003.
26. Shen, C., F.D. Vernier, C. Forlines and M. Ringel. Diamondspin: An Extensible Toolkit for Around-the-Table Interaction. *CHI*, 2004, 167-174.
27. Sousa, J.P. and D. Garlan. Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments. *IEEE Conf. on Software Architecture*, 2002.
28. Stefik, M., G. Foster, D.G. Bobrow, K. Kahn, S. Lanning and L. Suchman Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, *30* (1), 32-47.
29. Stewart, J., B.B. Bederson and A. Druin. Single Display Groupware: A Model for Co-Present Colloboration. *CHI*, 1999, 286-293.
30. Stoakley, C. and R. Pausch. Virtual Reality on a Wim: Interactive Worlds in Miniature. *CHI*, 1995, 265-272.
31. Streitz, N.A. and e. al. I-Land: An Interactive Landscape for Creativity and Innovation. *CHI*, 1999, 120-127.
32. Tan, D.S., B. Meyers and M. Czerwinski. Wincuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space. *CHI*, 2004. 1525-1528.
33. Tandler, P., et. al. Connectables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces. *UIST*, 2001, 11-20.
34. Tang, J.C. Findings from Observational Studies of Collaborative Work. *International Journal of Man-Machine Studies*, *34*, 143-160.
35. VanGundy, A.B. *Techniques of Structured Problem Solving*. Von Nostrand Reinhold Company, New York, NY, 1981.
36. Wickens, C.D. Multiple Resources and Performance Prediction. *Theoretical Issues in Ergonomic Science*, *3* (2), 159-177.
37. Xia, S., D. Sun, C. Sun, D. Chen and H. Shen. Leveraging Single-User Applications for Multi-User Collaboration: The CoWord Approach. *CSCW*, 2004, 162-171.