Evaluating Interaction Techniques in an Interactive Workspace: Comparing the Effectiveness of a Textual Interface, Virtual Paths Interface, and ARIS

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

ABSTRACT
ARIS is an interface that enables users to visually relocate applications and redirect input among myriad devices in an interactive workspace. While we previously claimed that ARIS is more effective than other interfaces for performing these tasks, this work seeks to empirically validate our claim. We compared the use of ARIS to an interaction design of a text-based and virtual paths interface for relocating applications and redirecting input in an interactive workspace. Results show that (i) users can relocate applications and redirect input faster with ARIS than a text-based interface, (ii) users commit fewer errors with ARIS than a text-based interface, (iii) users experience less workload and are more satisfied with ARIS than a text-based interface, and (iv) ARIS was comparable to the use of a virtual paths interface. ARIS is more effective than an interaction design that requires a user to mentally map and select textual identifiers, while supporting functionality beyond that of a virtual paths interface.

Author Keywords
Application relocation, Input redirection, Interactive workspace, Ubiquitous computing, Window manager.

ACM Classification Keywords
H.5.2 User Interfaces, H.5.3.b Collaborative Computing, C.3.h Ubiquitous Computing.

INTRODUCTION
To work productively in an interactive workspace [8], users need an effective interface for quickly and easily relocating applications and redirecting input among screens in the workspace. When brainstorming, for example, users need to rapidly spread alternative ideas across screens as they come in and out of favor, and redirect input to sketch or annotate those ideas. If the interface cannot support, or otherwise disrupts rapid exchange of diverse ideas, it will hinder rather than facilitate productive collaboration or individual work. We use the term *screen* to refer to a device such as a PDA, laptop, desktop, or plasma display, each driven by an independent, but networked system.

The interaction design of a text-based interface, e.g., [16] and virtual paths interface, e.g., [9, 13] allows application relocation and input redirection in an interactive workspace. In a text-based interface, a user must learn and recall how textual identifiers, e.g., device names, map or relate to physical screens or applications, which becomes more difficult as devices are added or removed from the space.

In a virtual paths interface, a user moves the cursor seamlessly (and an application, if implemented) to another screen. While this offers the perception of a single, shared workspace, effective use requires learning a mental model of how the screens connect, which is difficult if their layout changes often or does not afford intuitive connections.

To overcome limitations of these interaction designs, we developed an interface called ARIS [2]. ARIS provides an iconic representation of the physical workspace in a 2-D, fold-out view. Leveraging recognition over recall [11] and spatial memory, users interact with iconic representations of applications and screens to perform application relocation and input redirection tasks in the workspace.

To evaluate the efficacy of ARIS, we compared the use of ARIS to an interaction design of a text-based interface and a virtual paths interface for relocating applications and redirecting input in an interactive workspace. As the use of interactive workspaces is increasing for collaborative and individual work, our study seeks to better understand how alternative interface designs affect how well users can perform these central tasks, leading to lessons about how to develop more effective interfaces for interactive spaces.

Three large and two small screens in our workspace were labeled with a particular category of image content. Several images were placed in a PowerPoint application, one image per slide. A user viewed the image on a slide, decided to which category it belonged, and relocated the application to the matching screen. The user then redirected input to the local screen, typed an annotation for the image, redirected input back to the previous screen, advanced the slide, and repeated. A user performed four tasks with each interface. We measured performance, errors, subjective workload, and satisfaction. Sixteen users participated in the study.

Results show that (i) users can relocate applications and redirect input faster with ARIS than a text-based interface,

(ii) users commit fewer errors with ARIS than a text-based interface, (iii) users experience less workload and are more satisfied with ARIS than a text-based interface, and (iv) ARIS was comparable to a virtual paths interface. ARIS is more effective than an interaction design that requires a user to mentally map and select textual identifiers, while supporting functionality beyond a virtual paths interface.

RELATED WORK

In this section, we describe interactive workspaces, mechanisms for performing application relocation and input redirection tasks, and evaluations of those mechanisms.

Interactive Workspaces

An interactive workspace is a technology-rich, physical space that affords seamless sharing of digital information, dramatically improving collaborative, or even individual, activities for design, education, urban planning, and more [8]. The workspaces enable users to spread a large amount of digital information across multiple screens and interact with that information for collaborative or individual work.

Interactive workspaces are equipped with public or shared devices such as large screens, interactive tables, and graphics tablets. Users can also bring in their own portable devices such as PDAs, laptops, or Tablet PCs and connect them to the infrastructure managing the workspace.

Independent of the work domain, users need to perform two central tasks in these workspaces: *relocating applications* to spread information across screens and share it with others and *redirecting input* to interact with that information.

To support these tasks at a systems level, researchers have developed scalable distributed systems such as Gaia [14], iROS [7] and Aura [19] that enable multiple, independent devices to work together to form a single, larger system. However, effective interfaces are needed that enable users to quickly and easily perform application relocation and input redirection tasks, enabling them to work productively.

In our work, we conducted the first study comparing alternative interaction designs for performing relocation and redirection tasks, and show that ARIS is effective for performing these central tasks in an interactive workspace.

Mechanisms for Performing Application Relocation and Input Redirection Tasks

Researchers have developed many mechanisms to relocate applications and redirect input across multiple, independent machines and operating systems. We describe several such mechanisms and how they motivated the selection of the interaction designs used in our study.

In [10], researchers extended a web browser to enable users to browse web pages across multiple screens connected to independent machines. To relocate a browser window, a user selects the textual identifier of the destination screen from a list of available screens.

I-Land [20] has several novel interactions such as shuffle, throw, take, and pick-and-drop for relocating applications within large screens and between other screens.

In Easy Living [4], the managing infrastructure relocates applications automatically to devices nearest to the sensed presence of a user as the user moves about the environment.

With UbiTable [17] and augmented surfaces [13], users can share applications on a horizontal, interactive work surface using an interface that consists of iconic portals or virtual paths for relocating applications and redirecting input.

In Mighty Mouse [3], users redirect input by selecting the destination screen from a list of identifying icons. To end input redirection, the user performs a special click and key combination. PointRight [9] uses configurable geometric paths to enable users to redirect input across devices in an interactive workspace. This allows the user to move the cursor seamlessly (without a UI widget) across devices.

In iCrafter [12], a user relocates an application by relocating - or migrating - the service that supports it to another device. Using an interface that provides a top-down view of the workspace, the user drags a textual identifier of the service and drops it onto the destination screen. In contrast, ARIS enables users to perform application relocation and input redirection by interacting with iconic representations of applications and screens in a 2-D, *fold-out* view of the physical workspace [2]. ARIS is one of the few interfaces that integrate application relocation and input redirection into a single visual interface.

While we could not implement and test each of these interfaces in our study, our goal was to compare alternative interaction designs that typify many of those used in prior work. The text-based interface used in our study typifies interaction designs in iROS [7], Gaia [14], and Mighty Mouse [3], where part of the interaction is to mentally map and select textual identifiers for applications or physical screens in the environment. The virtual paths interface was used because it typifies interaction designs in UbiTable [17], augmented work surfaces [13], PointRight [9], and the extension of multi-monitor desktops to an interactive workspace. Our study is the first to compare alternative interaction designs for interactive workspace tasks.

Evaluations of the Mechanisms

Although usability studies of interfaces for relocating applications or redirecting input have been conducted [9, 10, 18], empirical comparisons of alternative interfaces for relocating applications and redirecting input have not. For example, Johanson et al. [9] evaluated the usability of PointRight for different users and task domains. They did not, however, compare the use of PointRight to other interaction designs such as Mighty Mouse or I-Land.

While evaluating the usability of an interface for relocating applications and redirecting input is important, our evaluation is the first to empirically compare *alternative* interaction designs for performing these tasks. The results of our study can help lead to more effective interfaces, making the use of interactive workspaces more productive.

USER STUDY

The purpose of our study was to compare alternative interfaces for relocating applications and redirecting input in an interactive workspace. Specifically, we designed our study to answer the following questions:

- How much does the interface affect how quickly a user can relocate applications and redirect input among screens in an interactive workspace?
- How much does the interface affect how many errors a user commits when performing those tasks?
- How much does the interface affect subjective workload when performing those tasks?
- How much does the interface affect user satisfaction when performing those tasks?

Experimental Design

The experiment used a doubly multivariate, repeated measures design with Interface (virtual paths interface, ARIS, and text-based interface) and Trial (relocation and redirection for each image in the application) as the factors.

Users

Sixteen users (7 female) participated in the study. Users consisted of undergraduate and graduate students, and administrative professionals from our institution. Ages ranged from 18 to over 40. Users were not compensated for participating in the study.

Hardware and Software

A high-end Dell Precision 450n workstation running Windows XP Professional was used to drive the five screens in the workspace. The workstation was equipped with one nVidia Quadro 1000 and two nVidia FX 5200 graphics cards. Camtasia was used to video record a user's screen interaction and a video camera was used to record a user's head movements.

Interactive Workspace

As shown in Figure 1, our interactive workspace consisted of three 61" plasma screens mounted on moveable stands and two 20" LCD screens. The LCD screens were positioned 2' apart on a table in the center of the room, faced in the same direction, and had resolution 1280x1024. We positioned two plasma screens behind the table directly in a user's field of view and physically close together along the same plane. The third plasma screen was positioned just to the left of the table, turned 90 degrees but still within a user's field of view. Their resolution was set to 1360x768.

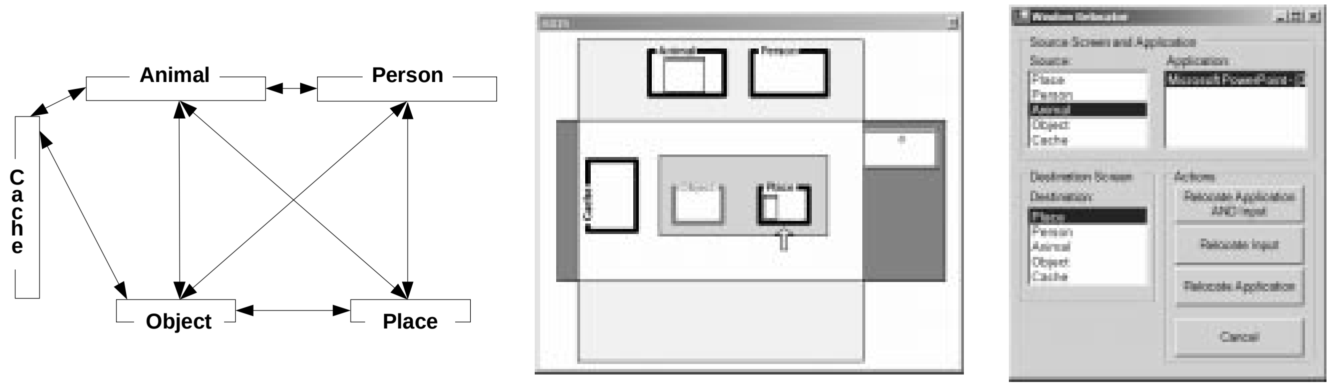
Figure 1: The interactive workspace used in our study. It consisted of two 20" LCD screens and three 61" plasma screens. Each screen had a label with a category name attached along the top that was used in the experimental task. A user viewed the current image in the application, relocated it to the screen with the appropriate label, redirected input to the local machine, typed an annotation, and redirected input back the screen with the application.

This configuration of screens is representative of existing interactive workspaces, e.g., [7, 14].

To conduct our study without bias across the interfaces, particularly for the virtual paths interface, we did not change the location of devices during the study (thus eliminating the need to reconfigure the virtual paths interface), we only used a mouse input device (as this was the only input device that could be effectively used across all three interfaces), and we configured the screens such that they were all within a user's field of view.

Because the distributed infrastructure used in interactive workspaces are research prototypes, we did not want slow response times or other errors in the underlying system to adversely affect a user's task performance or perception of an interface. Most importantly, we had to overcome the fact that no existing infrastructure for an interactive workspace supports application relocation in a manner consistent with a virtual paths interface. This is because providing an interaction where an application appears to move smoothly between two screens connected to independent machines is difficult. However, because this interaction design *could* be built, we wanted to include it in our comparative study.

To overcome these challenges, we *simulated* the distributed functionality of an interactive workspace through the use of a single, high-end PC with three multi-head graphics cards. Because the workspace was now just a single large desktop spread across five screens, a virtual paths interaction design could now be used to seamlessly relocate applications and redirect input. For ARIS and the interaction design of the text-based interface, application relocation and input redirection were achieved by setting the XY values of an



(a) The virtual path interface, arrows show the connections between screens. **(b) The ARIS interface.** **(c) The text-based interface.**

Figure 2: The three interfaces that were compared in our study.

application or the cursor to the appropriate location. This caused the application or cursor to immediately appear at that location and avoided any performance issues. Because we recorded the frame buffer using Camtasia, we could use the time stamps in the screen interaction video to measure performance, and thus did not have to instrument each interface separately.

Interfaces

In the user study, we compared three interfaces:

- *A virtual paths interface.* To perform a relocation task, a user selected the title bar of an application and dragged it to the desired screen. To redirect input, a user moved the cursor directly to the desired screen. Because we *simulated* the distributed functionality of the workspace, this interaction design provided a seamless method for performing application relocation and input redirection tasks. To configure the virtual connections, we conducted a pilot study where we asked three users to draw on paper how an application or the cursor should traverse the screens as it moved off each edge of a screen. Using the built-in Windows XP desktop controls, we configured the connections between screens based on these recommendations, which are shown in Figure 2a.
- *ARIS.* As shown in Figure 2b, ARIS provides an iconic representation of the applications and physical screens in an interactive workspace [2]. Configured in an XML file, the representation provides a 2-D, fold-out view of the physical workspace and the arrow indicates the user's location. To perform a relocation task, a user selects the iconic representation of the application and drags it to the desired screen in the representation. As the user drags the iconic representation, ARIS draws and updates a rectangular outline across the screens in the physical workspace to give feedback of the ongoing interaction. To redirect input, a user moves the cursor to the desired screen in the iconic representation and right-clicks. The right-click disambiguates the initiation of application relocation from redirection of input. The interface in ARIS went through extensive iterative prototyping, and the functional interface went through one round of usability evaluation. While an implementation exists for Gaia [15], an infrastructure for an interactive workspace, it was adapted to run in our simulated workspace.
- *A text-based interface.* As shown in Figure 2c, to perform a relocation task, a user selects the source screen, the application to move, the destination screen, and then clicks "relocate application." To redirect input, a user selects the destination screen and then clicks "relocate input." All selections are made from lists of textual identifiers. The identifiers in the lists matched the names on the labels attached to the physical screens. These were placed on the screens as part of the experimental task, discussed in the next section. Prior to our study, we evaluated the usability of the interface and refined the interaction design as necessary.

When using the text-based interface or ARIS in the study, multiple instances of the interfaces were created and located in the lower right corner of each screen. This was done to limit the interaction overhead of having to repeatedly access the interfaces from a menu or other control. Our experience with interactive workspaces also suggests that this is similar to how the interfaces would be configured in practice.

The interfaces compared in this study typify interaction designs used in existing interactive workspaces or other environments with similar goals. The text-based interface was used because it typifies interaction designs in iROS [7], Gaia [14], and Mighty Mouse [3], where at least part of the interaction is to select identifiers of applications or screens from lists and map them to their corresponding applications or physical screens in the environment.

The virtual paths interface was used because it typifies interaction designs where a user relocates applications and redirects input directly among screens, consistent with

UbiTable [17], augmented work surfaces [13], PointRight [9], and the natural extension of multi-monitor desktops to an interactive workspace.

We selected ARIS because we previously claimed that it would be more effective than a text-based or virtual paths interaction design in an interactive workspace. We made this claim because ARIS supports recognition over recall and enables a user to utilize their spatial memory when relocating applications and redirecting input.

Because we evaluated the usability of each interface and refined it prior to the study, any differences in the results from our study should be due to the interaction inherent in the interface and not due to poor usability of an interface.

Experimental Task

We wanted to develop a representative task that required a user to relocate applications and redirect input among screens in rapid succession. We felt this was necessary to stress the use of the interfaces for performance, and elicit meaningful workload and satisfaction responses from users.

Inspired by the use of the workspace for media annotation, the task was to relocate a PowerPoint application consisting of a sequence of images among screens in the workspace and to redirect input to the local screen to enter annotations. As shown in Figure 1, four screens were labeled with a category of image content, Person, Place, Animal, or Object, while the leftmost large screen was labeled Cache.

The PowerPoint application consisted of four images, one image per slide. A user viewed the image on a slide, relocated the application to the screen labeled with the category that fit that image (e.g., an image with a person in it went to the screen labeled Person), redirected input back to the local screen, typed an annotation for it (e.g., who it was), and then redirected input back to the screen with the application. These steps were repeated three more times, as there were four images in the application. The application always started on the leftmost large screen labeled Cache.

To control for how far a user had to move the application, we selected image sequences such that a user would relocate the application to each screen in the workspace exactly once, but always in a different order. Further control was provided by having a user locate the application within a rectangle drawn in the center of a screen (see Figure 1). This also provided a stopping goal for the user. Because three interfaces were being compared, three image sequences were created, plus two more for practice tasks.

The task was representative since a user had to relocate an application among screens based on its content and redirect input for local annotation. While application relocation and input redirection tasks are often performed in context of a collaborative activity, the tasks themselves are performed by an individual, thus we chose not to make the experimental task involve groups of users.

Procedure

Upon arriving at the lab, we went through an informed consent process with the user. The user filled out a demographic questionnaire and the experimenter described the equipment in the room and explained the task. The first interface was setup and described to the user. The user used the interface to perform a practice task consisting of six images (trials). If requested, a user could perform a second practice to ensure they understood the interface and task. Once questions were answered, the user performed the experimental task with the interface and was instructed to perform the task as quickly and accurately as possible.

Once finished, the user completed a NASA TLX and a post-task questionnaire while the next interface was setup. This process was repeated two more times. The ordering of the interfaces followed a Latin Square and the application for each interface was randomly assigned from the prepared set without replacement. After the final task, the user completed a post evaluation questionnaire where s/he ranked the use of the interfaces for performing the tasks. Camtasia was used to record a user's screen interaction and a video camera was used to record a user's physical head movements. The study lasted about one hour.

Measurements

In our study, we measured:

- *Time to relocate an application from one screen to another*. Relocation time was measured from when a user advanced the slide in the application to when the application appeared within the rectangle on the target screen. Measurements were computed from analysis of the time stamps in the screen interaction videos.
- *Time to redirect input from one screen to another*. Input redirection time was the time to redirect the cursor back to the local screen to enter the annotation and then to redirect the cursor back to the screen with the application. Because time to enter the annotation was not included, redirection time was computed in two parts. The first part was measured from when the cursor first moved in the direction of performing the redirection task to when the annotation window gained focus. The second part was measured from when the cursor just exited the annotation window to when the slide was advanced. Measurements were computed from analysis of the time stamps in the screen interaction videos.
- *Errors when relocating an application or redirecting input*. An error was defined to be any interaction step that did not move a user closer to completing the task. Example errors were misgrabbing the title bar with the virtual paths interface, moving the application to the wrong screen with ARIS, or selecting the wrong target screen and having to correct it with the text-based

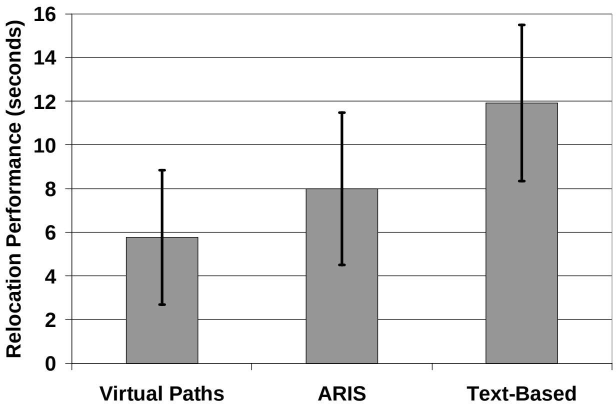


Figure 3: Mean performance time for application relocation.

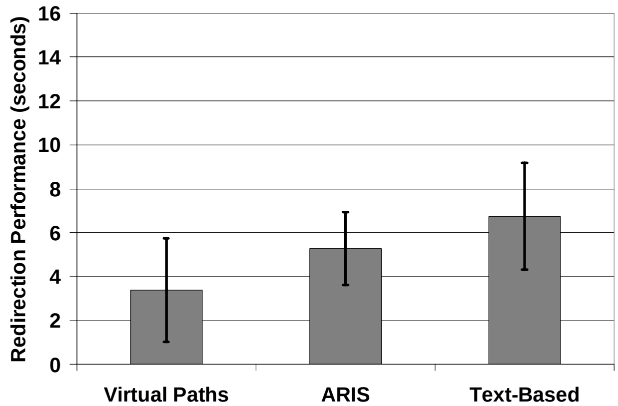


Figure 4: Mean performance time for input redirection.

interface. We refined a coding agenda [1], and used the agenda to code errors from the interaction videos.

- *Subjective workload*. This was measured using the NASA TLX [6]. The TLX measures workload along continuous scales in six dimensions: *mental demand, physical demand, temporal demand, own performance, effort*, and *frustration.* A user responds by marking a vertical line along a continuous scale from Low to High for each dimension. A mark was measured in 1/16" segments from the beginning of a scale.
- *User satisfaction.* Users rated and ranked an interface according to ease of use, appropriateness for the tasks, and ease of learning. A rating was structured using a 7-point Likert scale where statements were made in neutral form, e.g., the interface was easy to use, and users responded from 1 (Strongly Disagree) to 7 (Strongly Agree). Ranking responses were structured using a matrix where a user marked an 'X' in the most appropriate cell. A user ranked the interfaces from Best (1) to Worst (3) for each dimension. Users *rated* each interface immediately after performing the experimental task, while users *ranked* the interfaces after having completed the last experimental task, i.e., in a retrospective manner.

RESULTS

We discuss how the interfaces affected performance for application relocation and input redirection, error rate, subjective workload, and user satisfaction. Because Trial did not affect the measures, or interact with Interface, we report only on the main effects of Interface here.

Task Performance

Figure 3 shows a chart of user performance for relocating applications. An ANOVA showed that Interface had a main effect on how quickly a user could relocate an application ($F(2, 45)=11.71$, $p<0.001$). Post hoc analysis showed that a user relocated an application faster with both ARIS ($\mu=7.99$s) and the virtual paths interface ($\mu=5.75$s) than the text-based interface ($\mu=11.90$s, $p<0.045$, $p<0.002$, respectively). There was no difference between ARIS and the virtual paths interface. Because the interaction in the text-based interface typifies those often used for performing relocation tasks in existing interactive workspaces, the use of ARIS provides a meaningful performance improvement (~33%) for these tasks.

User performance for redirecting input is shown in Figure 4. An ANOVA showed that Interface had a main effect on how quickly a user could redirect input ($F(2, 45)=27.00$, $p<0.001$). Post hoc analysis showed that a user redirected input faster with the virtual paths interface ($\mu=3.39$s) than with both ARIS ($\mu=5.28$s, $p<0.003$) and the text-based interface ($\mu=6.74$s, $p<0.001$). A user redirected input faster with ARIS than a text-based interface ($p<0.018$).

The slower performance of ARIS relative to the virtual paths interface may have been partly due to the right-click interaction to redirect input, which caused some error delay for a few users. While we conducted a pilot study, this slight usability issue only surfaced in the experiment.

Overall, users were able to perform tasks with ARIS faster than with the text-based interface and nearly as fast as with the virtual paths interface.

Interface	Application Relocation	Input Redirection
ARIS	1 / 48 (2.08%)	3 / 48 (6.25%)
Virtual paths	5 / 48 (10.4%)	0 / 48 (0.00%)
Text-Based	13 / 48 (27.08%)	1 / 48 (2.08%)

Table 1: Task Errors Committed with Each Interface

Error Rate

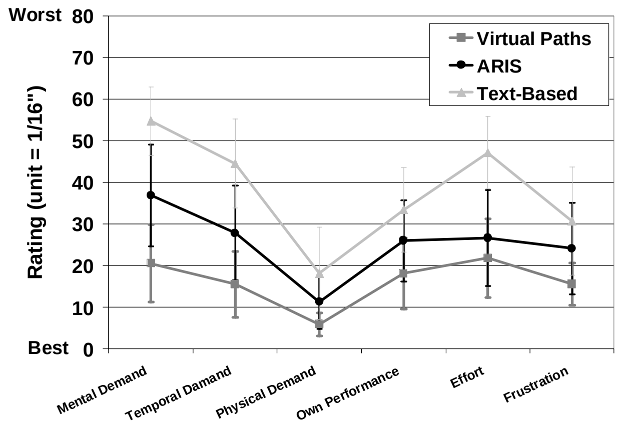


Figure 5: Ratings of subjective workload.

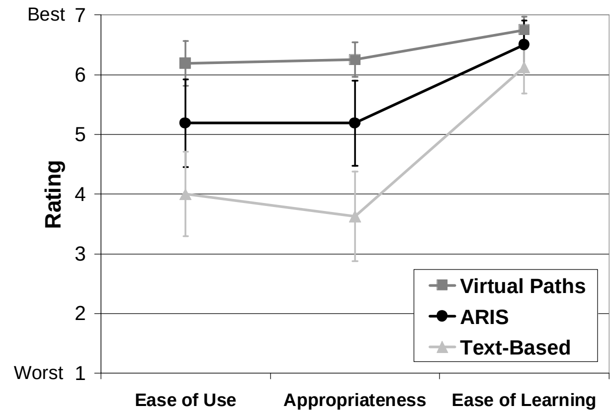


Figure 6: Post-task ratings for satisfaction.

Table 1 shows the errors committed with each interface. An ANOVA showed that Interface had a main effect on errors committed when relocating applications ($F(2, 30)=6.176$, $p<0.006$). Post hoc analysis showed that users committed fewer errors with ARIS ($\mu=0.063$) than with the text-based interface ($\mu=0.813$, $p<0.017$) when relocating applications. There were no other differences. An ANOVA showed that Interface did not affect errors committed when performing input redirection tasks ($F(2, 30)=1.84$, $p<0.176$).

Across tasks, users committed fewer total errors with ARIS than with the text-based interface (~71%), and the total number of errors committed with ARIS was low overall.

Subjective Workload

Figure 5 shows the ratings of subjective workload. A multivariate ANOVA showed that Interface had a main effect on subjective workload (Wilks' Λ = 0.277, $F(12, 50)=3.748$, $p<0.001$). Univariate tests showed that Interface affected mental demand ($F(2, 30)=27.46$, $p<0.001$), effort ($F(2, 30)=12.33$, $p<0.001$), temporal demand ($F(2, 30)=13.19$, $p<0.001$), physical demand ($F(2, 30)=4.99$, $p<0.028$), and frustration ($F(2, 30)=3.33$, $p<0.049$).

Post hoc analysis showed that the virtual paths interface had better ratings for mental demand ($p<0.001$), temporal demand ($p<0.001$), own performance ($p<0.036$), and effort ($p<0.004$) relative to the text-based interface. Relative to ARIS, the virtual paths interface had better ratings only for mental demand ($p<0.004$). This difference in mental demand is most likely due to the level of indirection inherent in the iconic representation used in ARIS.

ARIS had significantly better ratings for mental demand ($p<0.010$) and effort ($p<0.001$) compared to the text-based interface, and the trends were in the favorable direction along the other four dimensions of workload.

User Satisfaction

Figure 6 shows the ratings of user satisfaction for ease of use, appropriateness, and ease of learning. An ANOVA showed that Interface had a main effect on ratings of ease of use ($F(2, 30)=14.78$, $p<0.001$), appropriateness ($F(2, 30)=22.70$, $p<0.001$), and ease of learning ($F(2, 30)=6.95$, $p<0.003$). Post hoc analysis showed that ratings for ease of use, appropriateness, and ease of learning were higher for the virtual paths interface (μ=6.19, 6.25, 6.75, respectively) than the text-based interface (μ=4.00, 3.63, 6.13 with $p<0.001$, $p<0.001$, $p<0.010$, respectively). Post hoc analysis also showed that ratings for ease of use and appropriateness were higher for ARIS (μ=5.19, 5.19, respectively) than the text-based interface ($p<0.024$, $p<0.002$, respectively). The virtual paths interface was rated as more appropriate than ARIS ($p<0.049$) for performing the tasks.

Interface	1st (Best)	2nd	3rd
Ease of Use			
Virtual paths	10	5	1
ARIS	5	9	2
Text-based interface	1	2	13
Appropriateness			
Virtual paths	9	7	0
ARIS	6	8	2
Text-based interface	1	1	14
Ease of Learning			
Virtual paths	11	3	2
ARIS	2	11	3
Text-based interface	3	2	11

Table 2: Rank counts for each dimension of satisfaction.

Table 2 gives the post evaluation rankings of ease of use, appropriateness, and ease of learning, which are visually summarized in Figure 7. Interface affected users' rankings for each dimension of satisfaction (Pearson χ^2 (4, N=48) > 27.38, $p<0.001$ for each dimension). Across the dimensions,

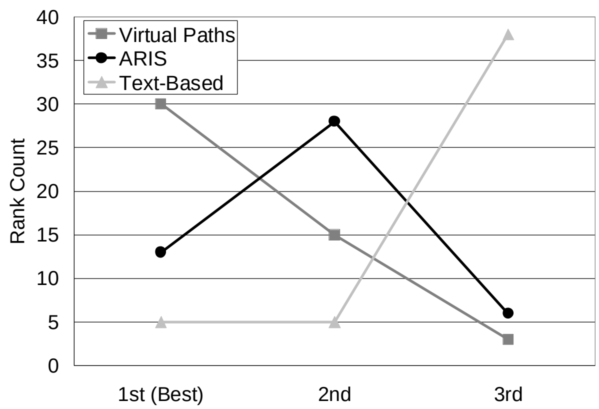


Figure 7: Combined post-evaluation rankings for satisfaction. ARIS was ranked much higher than the text-based interface, but not as highly as the virtual path interface.

users ranked ARIS much higher than the text-based interface, but not as highly as the virtual paths interface.

STRENGHTS AND WEAKNESSES OF ARIS

From the user study, we observed several strengths and weaknesses of ARIS. The strengths of ARIS were:

- *The use of an iconic representation of the workspace*. ARIS provides an iconic representation of the physical workspace in a 2-D, fold-out view. This enabled users to quickly associate the iconic representations in the interface with their corresponding applications and physical screens in the workspace. For example, one user commented "ARIS clearly depicts the whole space on one screen in a manner that is very accessible" and another commented "it's like the physical environment I am sitting in. So it's easier to correlate to the real environment and start where I left off."

- *Users could effectively use ARIS after just five minutes of instruction*. Before using ARIS in the study, we provided a brief overview of the interface and allowed users to practice using it for relocating applications and redirecting input, which lasted for about five minutes. Results show that users were then able to perform the experimental tasks quickly and with little error. This shows that novice users can quickly understand how to use ARIS. As one user stated "the mapping was very accurate and easy to think about when using."

- *ARIS provides continuous feedback of the ongoing interaction*. While a user drags an application's iconic representation across representations of screens in the interface, ARIS moves a rectangular live outline on the corresponding physical screen to provide feedback on where the application would be placed in the workspace. Analysis of the video recordings showed that users often glanced to the outline to check on or to complete the interaction, i.e., once the outline moved onto the destination screen, users would visually follow it to the desired location on that screen.

The weaknesses of ARIS were:

- *A level of indirection in the interface*. The iconic representation in ARIS has many strengths, but it also causes a user to work through a level of indirection. When performing a task in ARIS, a user must map the iconic representations in the interface to corresponding applications and physical screens in the workspace. While mapping the iconic representations in a spatial representation of the physical workspace requires less effort than mapping textual identifiers, a user must still work through a level of indirection.

- *The use of a right click to redirect input and the non-coupling of input redirection with application relocation*. To disambiguate input redirection from the start of application relocation (left click, then drag), we used a right-click for redirecting input in ARIS. Analysis of the screen interaction videos showed that a few users left clicked several times before recalling that a right-click was needed to redirect input in ARIS. A few users also commented that the "right click is confusing". While input redirection is independent of application relocation in ARIS, a few users felt that input redirection should be coupled with application relocation. One user said "it is unintuitive that the cursor does not follow the application after a drag and drop." We intend to resolve these issues in future work.

LESSONS LEARNED

From the study, we learned that:

- *ARIS enabled users to relocate applications faster than a text-based interface and was comparable to a virtual paths interface*. Our results show that users performed relocation tasks with ARIS about 33% faster than with the text-based interface and about as fast as with the virtual paths interface.

- *ARIS caused users to commit fewer errors than a text-based interface and was comparable to a virtual paths interface.* Our results show that when using ARIS users committed about 71% fewer errors across all tasks than when using the text-based interface and about as many as when using the virtual paths interface. We attribute the lower error rate relative to the text-based interface in part to the continual feedback provided by the live outlines drawn by ARIS when relocating applications.

- *ARIS induced less subjective workload on users than a text-based interface and was comparable to a virtual paths interface.* ARIS induced about 34% less workload than the text-based interface. While ARIS induced more mental demand than the virtual paths interface, they were comparable on the other five dimensions of workload. The increase in mental demand can most likely be attributed to the level of indirection inherent in the iconic representation in ARIS. However, as users gain more experience with ARIS, the mental demand imposed by its interface should decrease. As one user stated "I think ARIS will take longer to get used to but after some experience will be the easiest [to use]."

- *ARIS enabled users to redirect input faster than a text-based interface and was more satisfying to use than a text-based interface.* Our results show that users redirected input with ARIS about 22% faster than with the text-based interface. Users rated and ranked the use of ARIS higher along each dimension of satisfaction than the text-based interface. We believe that users were more satisfied with ARIS because of its spatial representation. For example, users stated "screens arranged in terms of physical layout was helpful" and "the map was nicely laid out and easy to understand."
- *A virtual paths interface enabled users to redirect input faster than ARIS and was ranked higher than ARIS for satisfaction.* The virtual paths interface enabled users to redirect input about 36% faster than with ARIS. This difference can most likely be attributed to the level of indirection in ARIS, the usability issue of the right-click to redirect input, and that ARIS requires a precise motor movement while the virtual paths interface allowed a ballistic motor movement. For the virtual paths interface, we observed that users often performed rapid 'flicks' of the wrist to roughly position the cursor on the target screen and then used precise movements for exact positioning. While users *ranked* the virtual paths interface as being the most satisfying interface, they *rated* the use of ARIS as highly as the virtual paths interface for most measures of satisfaction.
- *ARIS is more effective than an interaction design that requires a user to mentally map and select textual identifiers. It does this while supporting functionality beyond a virtual paths interface with little additional overhead.* Relative to the text-based interface, ARIS enabled users to perform tasks faster and commit fewer errors, induced less workload, and was much more satisfying to use. As one user stated, "I don't like the text-based interface because it forces me to remember which screen had which label." Because ARIS supports visual relocation and redirection, in contrast to an interaction design that requires mentally mapping and selecting textual identifiers, it is a more effective interface. Although the virtual paths interface was better on a few measures, ARIS was comparable for relocating applications, errors committed, five of six workload measures, and most ratings of satisfaction. However, because it uses an iconic, visual representation of the physical workspace, ARIS supports functionality beyond that of a virtual paths interface with little additional overhead.

Because the distributed functionality of an interactive workspace was *simulated* in our study, a virtual paths interaction design supported seamless relocation of applications and redirection of input. However, because a *realistic* interactive workspace is comprised of multiple, heterogeneous devices, implementing that same seamless interaction experience would be very difficult. With ARIS, the interaction experience in a realistic workspace is the same and an implementation exists [2].

Because ARIS uses an iconic representation of the physical workspace, it supports functionality beyond that of a virtual paths interface. For example, ARIS supports relocation of applications using a stylus input device, which would be awkward to perform using a virtual paths interface, as the stylus would have to be switched to a relative positioning mode. While difficult with a virtual paths interface, ARIS easily supports relocation of applications among screens whose viewable surfaces are not in the field of view, e.g., if users are sitting across from each other with their own local devices. Through real-time updates to its iconic representation, ARIS can immediately reflect the changing presence and location of portable devices, which would cause repeated reconfiguration of connections in a virtual paths interface, making it more difficult to learn and recall. Also, the iconic representation in ARIS could convey activity awareness, important for effective group work [5].

FUTURE WORK

For future work, we intend to:

- *Improve the usability of ARIS based on our results*. The comparative study highlighted a few usability issues with ARIS that were not previously identified. We intend to modify ARIS such that it will couple input redirection with application relocation unless otherwise specified by the user. We will also modify ARIS to support the use of a left click to redirect input in the interface.
- *Integrate a virtual paths interface for input redirection into ARIS.* In our study, we observed that users often performed ballistic movements (rapid flicks of the wrist) to redirect input with the virtual paths interface. Because input redirection toolkits exist for interactive workspaces [9], this mechanism could be integrated into ARIS. A user could thus use ballistic movements to redirect input while still using ARIS's iconic representation to relocate applications. We also want to test this integrated interface against a virtual paths interface in a similar study.
- *Support more group-based information and interaction in ARIS.* While ARIS shows application and cursor location information today, we want to enhance its interface to enable users to set and view access permissions for shared displays in the workspace, to identify specific applications as being "public" and then only show those

applications, and to view which applications other users are interacting with to better convey activity awareness.

CONCLUSION

To work productively in an interactive workspace, users need an effective interface for quickly and easily relocating applications and redirecting input among screens. We empirically compared a text-based interface, a virtual paths interface, and ARIS for performing relocation and redirection tasks in an interactive workspace. Results show that ARIS was more effective than an interaction design that requires a user to mentally map and select textual identifiers, while imposing little overhead beyond that of a virtual paths interface. Through its iconic representation, however, ARIS supports functionality beyond that of a virtual paths interface. Our work shows that the use of ARIS can enable users to work more productively in an interactive workspace than they can today.

REFERENCES

1. Altheide, D.L. *Qualitative Media Analysis*. Sage, Newbury Park, CA, 1996.
2. Biehl, J.T. and B.P. Bailey. ARIS: An Interface for Application Relocation in an Interactive Space. In *Proc. Graphics Interface*, London, Ontario, Canada, 2004, 107-116.
3. Booth, K.S., B.D. Fisher, C.J.R. Lin and R. Argue. The "Mighty Mouse" Multi-Screen Collaboration Tool. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2002, 209-212.
4. Brumitt, B., B. Meyers, J. Krumm, A. Kern and S.A. Shafer. Easyliving: Technologies for Intelligent Environments. In *Proc. Handheld and Ubiquitous Computing*, 2000, 12-29.
5. Gutwin, C. and S. Greenberg. Effects of Awareness Support on Groupware Usability. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1998, 511-518.
6. Hart, S.G. and L.E. Stateland. Development of Nasa-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Hancock, P.A. and Meshkati, N. (eds.) *Human Mental Workload*, North-Holland, Amsterdam, 1988, 139-183.
7. Johanson, B. and A. Fox. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2002.
8. Johanson, B., A. Fox and T. Winograd. Experience with Ubiquitous Computing Rooms. *IEEE Pervasive Computing*, 1, 2002, 67-74.
9. Johanson, B., G. Hutchins, T. Winograd and M. Stone. Pointright: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Paris, France, 2002, ACM, 227-234.
10. Johanson, B., S. Ponnekanti, C. Sengupta and A. Fox. Multibrowsing: Moving Web Content across Multiple Displays. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Seattle, Washington, 2001, 346-353.
11. 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, 1989.
12. Ponnekanti, S.R., B. Lee, A. Fox, P. Hanrahan and T. Winograd. Icrafter: A Service Framework for Ubiquitous Computing Environments. In *Proc. Ubiquitous Computing Conference*, 2001.
13. Rekimoto, J. and M. Saitoh. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Pittsburgh, PA, 1999, 378-385.
14. 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, 2002, 74-83.
15. Román, M., H. Ho and R. Campbell. Application Mobility in Active Spaces. In *International Conference on Mobile and Ubiquitous Multimedia*, 2002.
16. Schilit, B.N., N.I. Adams and R. Want. Context-Aware Computing Applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, IEEE Computer Society, 1994, 85-90.
17. Shen, C., K.M. Everitt and K. Ryall. Ubitable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. In *Proc. Ubiquitous Computing Conference*, 2003.
18. Shen, C., F.D. Vernier, C. Forlines and M. Ringel. Diamondspin: An Extensible Toolkit for around-the-Table Interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2004, 167-174.
19. Sousa, J.P. and D. Garlan. Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments. In *Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture*, 2002.
20. Streitz, N.A. and e. al. I-Land: An Interactive Landscape for Creativity and Innovation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1999, 120-127.