Comparing an Iconic Interface to a Text-based and Virtual Paths Interface for Effective Interaction in an Interactive Workspace

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

ARIS is an iconic interface that enables users to relocate applications and redirect input in an interactive workspace. We compared the use of ARIS to a text-based and virtual paths (VP) interface for relocating applications and redirecting input. Results showed that the iconic interface was more effective than the text-based interface and almost as effective as the VP interface. Since the use of text-based interfaces is common, the use of an iconic interface enables relocation and redirection tasks to be performed more effectively than today. Though it imposes a small additional cost, an iconic interface also supports features useful for interactive spaces that a VP interface does not.

Introduction

To work productively in an interactive workspace (Johanson, Fox, & Winograd, 2002), users need an effective interface for quickly and easily relocating applications and redirecting input among screens. To support effective brainstorming (VanGundy, 1981), for example, users must be able to spread alternative ideas across screens and redirect input to sketch or annotate those ideas. If the interface cannot support, or otherwise disrupts rapid exchange of ideas, it will hinder rather than facilitate individual or collaborative work. Our work seeks to better understand how three classes of interfaces; iconic, text, and virtual paths, support relocation and redirection tasks in an interactive space.

The interaction design of a text-based interface, e.g., (Román, Ho, & Campbell, 2002) and a virtual paths (VP) interface, e.g., (Johanson, Hutchins, Winograd, & Stone, 2002) are two common approaches for enabling application relocation and input redirection. However, when using a text-based interface, users must learn and recall how textual identifiers, e.g., device names or IP addresses, map to physical screens or applications, which becomes difficult as devices are added or removed from the workspace. In a VP 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. Also, since the use of virtual paths requires the cursor to be controlled beyond the local screen, it does not support stylus input, touch input, or other absolute positioning input devices.

To overcome limitations of these interfaces, we developed an interface called ARIS (Biehl & Bailey, 2004). ARIS provides an iconic, spatial representation of a physical workspace in a 2D, fold-out view. Leveraging recognition over recall (Johnson, 1989) and spatial memory, users are able to interact with iconic representations to perform relocation and redirection tasks. Since the tasks are fully performed from the interface on the local screen, ARIS supports the use of both relative (e.g., mouse) *and* absolute (e.g., stylus) input devices.

In this work, we empirically compare the use of ARIS to a text-based and VP interface for relocating applications and redirecting input, two of the most frequent tasks in an interactive workspace. These interfaces were selected because they represent commonly used interfaces for interactive workspaces. In our study, users were asked to complete several relocation and redirection tasks as part of a broader, meaningful activity. This activity consisted of users organizing a series of digital images across several screens and adding short annotations to each image.

Results show that an iconic interface is more effective than a text interface and almost as effective as a VP interface. Since text-based interfaces are common in interactive workspaces, an iconic interface enables relocation and redirection tasks to be performed more effectively than today. In addition, our results show that an iconic interface imposes only a small additional cost to the user beyond that of a VP interface. This is very encouraging because an iconic interface supports functionality useful for interactive workspaces that a VP interface cannot easily support.

Related Work

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

Interactive Workspaces

An interactive workspace is a technology-rich, physical space that affords seamless sharing of digital information, with the potential to dramatically improve individual and collaborative work for design, education, urban planning, and more (Johanson, Fox et al., 2002). Interactive workspaces contain public/shared devices such as large screens, interactive tables, and graphics tablets as well as private/portable devices such as PDAs, laptops, and Tablet PCs. Independent of the work domain, productive use of an interactive workspace requires users to effectively perform two common and frequent tasks: *relocating applications* to spread information across screens 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 (Román, Hess et al., 2002) and iROS (Johanson & Fox, 2002) 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 relocation and redirection tasks, facilitating more productive work.

Mechanisms for Performing Application Relocation and Input Redirection Tasks

In (Johanson, Ponnekanti, Sengupta, & Fox, 2001), 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 (Streitz & al., 1999) has several interactions such as shuffle, throw, take, and pick-and-drop for relocating applications within large screens and between other screens. In Easy Living (Brumitt, Meyers, Krumm, Kern, & Shafer, 2000), the managing infrastructure relocates applications automatically to devices nearest the sensed presence of a user as the user moves about the environment. With UbiTable (Shen, Everitt, & Ryall, 2003) users can share applications on a horizontal, interactive work surface using an interface that consists of iconic portals or virtual paths for managing applications and input.

In iCrafter (Ponnekanti, Lee, Fox, Hanrahan, & Winograd, 2001), a user relocates an application by 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 Mighty Mouse (Booth, Fisher, Lin, & Argue, 2002), users redirect input by selecting the destination screen from a list of identifying icons. PointRight (Johanson, Hutchins et al., 2002) uses geometric paths to enable input redirection, which allows the cursor to be moved seamlessly (without a UI widget) across devices.

In contrast to these mechanisms, ARIS enables users to perform relocation and redirection tasks by interacting with iconic representations of applications and screens in a 2D, *fold-out* view of the workspace (Biehl & Bailey, 2004). ARIS is one of the few interfaces that integrates relocation and redirection tasks into a single visual metaphor.

Evaluations of the Mechanisms

Although usability studies of interfaces for relocating applications or redirecting input have been conducted (Johanson, Hutchins et al., 2002; Johanson et al., 2001; Shen, Vernier, Forlines, & Ringel, 2004), empirical comparisons of alternative interfaces for relocating applications and redirecting input have not. For example, (Johanson, Hutchins et al., 2002) evaluated the usability of PointRight for different users and task domains. They did not, however, compare the use of PointRight to other interaction designs.

Though results related only to the iconic interface were previously reported in (Biehl & Bailey, 2005), this paper reports the complete set of empirical results comparing all three interfaces. This is a substantial extension to our prior work as it allows us to empirically understand differences between the interfaces and related implications.

User Study

We designed our study to answer these questions:

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

Experimental Design and Workspace Configuration

The experiment used a doubly multivariate, repeated measures design with Interface (VP, ARIS, and text-based) and Trial (relocation and redirection for each image) as the factors. Sixteen users participated in the study.

Figure 1: The workspace used in our study.

Our workspace consisted of 3 plasma screens and 2 LCD screens (Figure 1). The LCDs were positioned 2' apart on a table, faced in the same direction. We positioned 2 plasma screens behind the table directly in a user's field of view and physically close together along the same plane. The third screen was positioned to the left of the table, turned 90 degrees but still within a user's field of view. This configuration is representative of many existing workspaces, e.g., (Johanson, Fox et al., 2002; Román, Hess et al., 2002).

To conduct our study without bias across the interfaces, particularly for the VP interface, we did not change the location of devices during the study (eliminating the need to re-acquire a mental model of screen connections), we only used a mouse device (the only device that could be effectively used across all interfaces), and we physically configured the screens such that they were all within a user's field of view.

Because the infrastructures used to drive interactive workspaces are research prototypes, we did not want slow responses or other errors in the underlying system to adversely affect a user's performance or perception of an interface. Most importantly, we also had to overcome the fact that no existing infrastructure supports application relocation in a manner consistent with a VP interface. This is because providing an interaction where an application appears to move smoothly between two screens connected to independent machines is very difficult. However, because this interaction design *could* be built, we wanted to include it in our study. To overcome these challenges, we *simulated* the system functionality of an interactive workspace by using a single PC with multiple graphics cards.

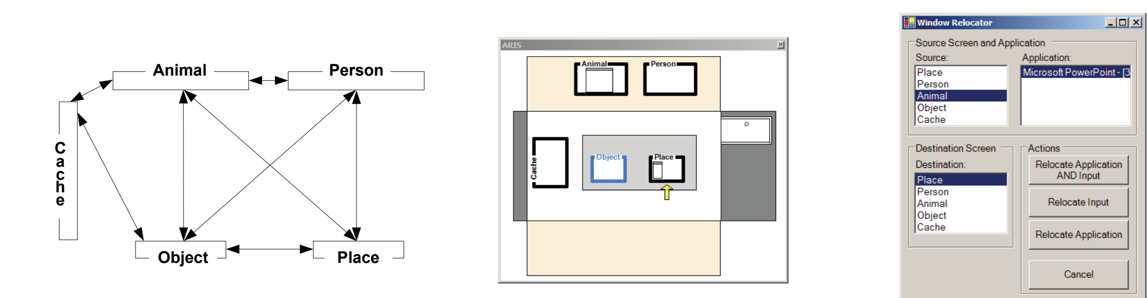
Interfaces

In the user study, we compared three interfaces:

- *A virtual paths interface*. To perform a relocation, 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. 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. The final configuration is shown in figure 2a.
- *ARIS*. As shown in figure 2b, ARIS provides an iconic representation of the applications and physical screens in a 2-D, fold-out view (Biehl & Bailey, 2004). To relocate an application, the user selects the iconic representation of the application and drags it to the desired screen in the interface. To redirect input, a user moves the cursor to the desired screen in the iconic representation and right clicks.
- *A text-based interface*. As shown in figure 2c, to perform a relocation task, a user selects the source screen, application, and destination screen; then selects "relocate application." To redirect input, a user selects the destination screen and selects "relocate input." All selections are made from lists of textual identifiers, which matched labels attached to the physical screens. The labels were attached to the screens as part of the experimental task.

For the text-based interface and ARIS, an instance of the interface was placed in the lower right corner of each screen. This was done to limit the overhead of having to repeatedly access the interfaces from a menu or other control. Our experience 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 workspaces. The text-based interface typifies interaction designs in iROS (Johanson & Fox, 2002) and in Gaia (Román, Hess et al., 2002), where at



(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.

least part of the interaction is to select identifiers of applications or screens from lists. The VP interface typifies interaction designs where users relocate applications and input directly among screens, consistent with UbiTable (Shen et al., 2003), PointRight (Johanson, Hutchins et al., 2002), and the natural extension of multi-monitor desktops. We selected ARIS because it typifies interaction designs in interfaces that use 2D spatial metaphors such as iCrafter (Ponnekanti et al., 2001). Finally, each interface went through at least one round of usability testing prior to the study. Thus, any differences should be due to the interaction design of the interfaces, not to poor usability.

Experimental Activity

The activity was to relocate a PowerPoint application among screens 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, and then redirected input back to the screen with the application. These steps were repeated three more times. The application always started on the large screen labeled Cache.

The activity required users to perform many relocation and redirection tasks 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. The activity 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 elected not to make the activity involve groups of users. This will be studied in future work.

Procedure & Measurements

When the user arrived at the lab, the first interface was setup and described, and the user performed some practice tasks. The user then performed the experimental activity and was instructed to work as quickly and accurately as possible. Once finished, the user completed a post-task questionnaire. This process was repeated two more times. The ordering of the interfaces followed a Latin Square and the images used for each condition were randomly assigned from the prepared set without replacement. After the last activity, the user completed a questionnaire where he ranked the interfaces for performing the tasks. Camtasia was used to record a user's screen interaction.

In our study, we measured: time to relocate an application from one screen to another; time to redirect input from one screen to another; errors when relocating an application or redirecting input; subjective workload, measured using the NASA TLX (Hart & Stateland, 1988); and user satisfaction.

Results

We discuss how the interfaces affected performance, number of errors, subjective workload, and satisfaction. Since Trial did not affect the measures, or interact with Interface, we report only the main effects of Interface.

Performance

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 (μ=7.99s) and the VP interface (μ=5.75s) than the text-based interface (μ=11.90s, $p<0.045$, $p<0.002$, respectively). There was no difference between ARIS and the VP interface. Relative to the text-based interface, the use of ARIS provides a meaningful performance improvement (~33%) for this task. Interface also 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 VP interface (μ=3.39s) than with both ARIS (μ=5.28s, $p<0.003$) and the text-based interface (μ=6.74s, $p<0.001$). A user redirected input faster with ARIS than the text interface ($p<0.018$). Overall, users were able to perform tasks with ARIS faster than with the text-based interface and nearly as fast as with the VP interface.

Error Rate

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 users committed fewer errors with ARIS (μ=0.063) than with the text-based interface (μ=0.813, $p<0.017$). There were no other differences. Interface did not affect errors committed when redirecting input ($F(2, 30)=1.84$, $p<0.176$). Overall, users committed about 71% fewer errors with ARIS than with the text-based interface, and the total number of errors with ARIS was low overall.

Subjective Workload

A MANOVA 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 VP interface had better ratings for mental demand ($p<0.001$), temporal demand ($p<0.001$), own performance ($p<0.036$), and effort ($p<0.004$) than the text-based interface. Relative to ARIS, the VP 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 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 other dimensions.

User Satisfaction

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 VP interface (μ=6.19, 6.25, 6.75, respectively) than the text-based interface (μ=4.00, 3.63, 6.13 with $p<0.001$, $p<0.001$, $p<0.010$). 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 VP interface was rated as more appropriate than ARIS ($p<0.049$) for performing the tasks. Interface affected users' rankings for each dimension of satisfaction (Pearson χ^2 (4, N=48) > 27.38, $p<0.001$). Across dimensions, users ranked ARIS higher than the text interface, but not as high as the VP interface.

Discussion

To summarize, our empirical results show that:

- *The iconic interface was more effective than the text-based interface.* When using ARIS, users relocated applications 33% faster, redirected input 22% faster, and committed 71% fewer errors relative to the text interface. ARIS induced 34% less workload and users rated and ranked it higher along each dimension of satisfaction. We believe ARIS performed better because it allowed users to leverage their spatial reasoning abilities to map between the iconic representation and physical workspace. For example, users stated "screens arranged in terms of physical layout was helpful" and "it's like the physical environment I am sitting in. So it's easier to correlate to the real environment."
- *The virtual paths interface was more effective than the iconic interface for some measures.* Compared to ARIS, the VP interface enabled users to redirect input 36% faster and induced 44% less mental demand. Users rated it 20% more appropriate for the task and 19% easier to use. While the VP interface performed better on these measures, other measures showed no difference. For example, no differences were found for relocation time, ease of learning, errors committed, and 5 of the 6 workload measures. The differences in mental demand and satisfaction may be a result of the users having to perform actions through a level of indirection introduced by the iconic interface. We believe these differences will narrow as users become more familiar interacting with iconic interfaces. As one user stated, "I think ARIS will take longer to get used to but after some experience will be the easiest [to use]."

Our results have two important implications. First, since text interfaces are commonly used in interactive spaces, the use of an iconic interface enables relocation and redirection tasks to be performed more effectively than today. Second, our results show that an iconic interface imposes only a small additional cost to the user beyond that of a VP interface. This is important since an iconic interface supports several useful features that a VP interface cannot. For example, an iconic interface also supports the use of stylus and touch input. This allows relocation tasks to be performed from Tablet PCs, Wacom Tablets, and SmartBoards; all of which are common in interactive workspaces.

The visual representation in an iconic interface provides advantages over both the VP and text interfaces. For example, it provides visual cues that can help users maintain awareness of which applications are on which screens, important for collaborative work (Gutwin & Greenberg, 2002). Also, these cues can be extracted with a quick glance.

Finally, recall that in our study a VP interface was simulated by using one computer to drive all the displays. This was necessary because a complete implementation of a VP interface that supports input redirection *and* application relocation does not currently exist and would be difficult to implement. Our results show that implementing such an interface may not actually be necessary. The reason is that users would gain only moderate performance improvements over iconic interfaces, which already exist, and iconic interfaces would still support additional functionality.

Conclusion

As the use of interactive workspaces is becoming increasingly common, it is important to understand how alternative interfaces affect two common tasks in these environments – relocating applications and redirecting input. This work has made important contributions in this direction by comparing the use of an iconic interface to a text-based and virtual paths interface for performing these tasks. Results showed that an iconic interface is more effective than a text-based interface and almost as effective as a virtual paths interface. Since text-based interfaces are commonly used in interactive workspaces, an iconic interface enables relocation and redirection tasks to be performed more effectively than today. Though an iconic interface imposes a small additional cost to the user, it supports additional functionality useful for interactive workspaces that a VP interface cannot easily support; e.g., the use of absolute input devices as well as visual cues that help convey awareness. As such, iconic interfaces offer an alternative and promising interaction metaphor for interactive workspaces.

References

Biehl, J. T., & Bailey, B. P. (2004). *ARIS: An Interface for Application Relocation in an Interactive Space.* Paper at Graphics Interface. pp.107-116.

Biehl, J.T., & Bailey, B.P. (2005). *A Toolset for Creating Iconic Interfaces for Interactive Workspaces.* Paper at the IFIP International Conference on Human-Computer Interaction (INTERACT). pp. 699-712.

Booth, K. S., Fisher, B. D., Lin, C. J. R., & Argue, R. (2002). *The "Mighty Mouse" Multi-Screen Collaboration Tool.* Paper at the ACM Symposium on User Interface Software & Technology. pp. 209-212.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., & Shafer, S. A. (2000). *EasyLiving: Technologies for Intelligent Environments.* Paper at the Conference on Handheld and Ubiquitous Computing. pp.12-29.

Gutwin, C. & Greenberg, S. (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Journal of Computer-Supported Cooperative Work* (3-4), 411-446.

Hart, S. G., & Stateland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of emperical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human Mental Workload* (pp. 139-183). Amsterdam: North-Holland.

Johanson, B., & Fox, A. (2002). *The Event Heap: A Coordination Infrastructure for Interactive Workspaces.* Paper at the 4th IEEE Workshop on Mobile Computing Systems and Applications. (WMCSA). pp. 83-93.

Johanson, B., Fox, A., & Winograd, T. (2002). The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing, 1*(2), 67-74.

Johanson, B., Hutchins, G., Winograd, T., & Stone, M. (2002). *PointRight: Experience with Flexible Input Redirection in Interactive Workspaces.* Paper at the ACM Symp. on User Interface Software & Technology. pp. 227-234.

Johanson, B., Ponnekanti, S., Sengupta, C., & Fox, A. (2001). *Multibrowsing: Moving Web Content Across Multiple Displays.* Paper at Ubiquitous Computing Conference (UBICOMP). pp. 346-353.

Johnson, J., T. L. Roberts, W. Verplank, D. C. Smith, C. H. Irby, M. Beard, and K. Mackey. (1989). The Xerox Star: a retrospective. *IEEE Computer, 22*(9), 11-26.

Ponnekanti, S. R., Lee, B., Fox, A., Hanrahan, P., & Winograd, T. (2001). *iCrafter: A Service Framework for Ubiquitous Computing Environments.* Paper at Ubiquitous Computing Conference (UBICOMP). pp. 56-75.

Román, M., Hess, C., Cerqueira, R., Ranganat, A., Campbell, R., & Nahrstedt, K. (2002). Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing, 1*(4), 74-83.

Román, M., Ho, H., & Campbell, R. (2002). *Application Mobility in Active Spaces.* Paper at the International Conference on Mobile and Ubiquitous Multimedia.

Shen, C., Everitt, K. M., & Ryall, K. (2003). *UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces.* Paper at the Ubiquitous Computing Conference (UBICOMP). pp. 281-288.

Shen, C., Vernier, F. D., Forlines, C., & Ringel, M. (2004). *DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction.* Paper at the ACM Conf. on Human Factors in Computing Systems. pp. 167-174.

Streitz, N. A., & al., e. (1999). *i-LAND: AN Interactive Landscape for Creativity and Innovation.* Paper at the ACM Conference on Human Factors in Computing Systems. pp. 120-127.

VanGundy, A. B. (1981). *Techniques of Structured Problem Solving*. New York, NY: Von Nostrand Reinhold Company.