A Toolset for Creating Iconic Interfaces for Interactive Workspaces

Jacob T. Biehl and Brian P. Bailey

Department of Computer Science, University of Illinois, Urbana, IL 61801
{jtbiehl, bpbailey}@uiuc.edu

Abstract. To work productively in an interactive workspace, users need effective interfaces for seamlessly sharing, annotating, and juxtaposing digital information across heterogeneous devices. In this paper, we present an interface toolset for constructing and using iconic interfaces for interactive workspaces. Using an iconic representation of the physical workspace, users can quickly and easily relocate applications and redirect input across devices. The toolset provides a graphical tool for rapidly constructing iconic representations for various workspaces, supports an existing interactive workspace infrastructure, and is engineered to be portable to others. A usability evaluation showed that the interaction design of the interfaces created with our toolset is effective for redirection and relocation tasks. Our results provide the first empirical baseline for comparing alternative interfaces for interactive workspaces. The use of our toolset facilitates more productive use of interactive workspaces for both individual and group work and is available for download today.

1 Introduction

An interactive workspace is a physical workspace that connects small and portable devices, and large shared displays through a distributed software infrastructure. Because they enable information to be easily distributed and synchronized across devices, interactive workspaces can dramatically improve how users share, annotate, and juxtapose digital information for individual and group work [7].

Two of the most common and frequent tasks in an interactive workspace are relocating applications and redirecting input across devices. While systems-level services for performing these tasks are fully supported within current distributed infrastructures for interactive workspaces, users need effective interfaces to quickly and easily perform those tasks. To support application relocation and input redirection tasks, several interfaces and supporting toolsets have been developed, e.g., [6, 15, 18]. However, these interfaces are either too heavily tied to a specific infrastructure, making them difficult to port to other infrastructures, or variations of the interfaces are overly difficult to construct for different workspaces. Additionally, for many of these interfaces, the effectiveness of their interaction design has not been evaluated.

In our previous work, we discussed the iterative design of an interface which uses an iconic representation for performing application relocation and input redirection tasks [1]. The iconic representation supports recognition over recall and enables users to utilize their spatial reasoning abilities when performing these tasks. In this work,

M.F. Costabile and F. Paternò (Eds.): INTERACT 2005, LNCS 3585, pp. 699 – 712, 2005.

we present a toolset for constructing iconic interfaces for interactive workspaces. We explain how iconic representations are constructed and used, describe the architecture and implementation of the toolset, and discuss results from a usability evaluation.

Our toolset provides a configuration tool that enables users to rapidly construct iconic representations of various physical workspaces. We implemented our toolset to work with an existing infrastructure, Gaia, but engineered it to allow easy integration with other existing infrastructures. The usability evaluation showed that the interaction design of our interface is effective for performing input redirection and application relocation tasks quickly and with minimal error, induces low subjective workload, and supports high satisfaction across users. Moreover, our results provide the first empirical baseline for comparing alternative interfaces for performing these tasks in interactive workspaces. By facilitating the creation of effective interfaces, our toolset enables more productive use of interactive workspaces for both individual and group work and is available for download today.

2 Related Work

In this section, we discuss how our work builds upon distributed infrastructures for interactive workspaces, differs from other interaction designs for performing application relocation and input redirection tasks, and compares to other toolsets for creating and executing user interfaces for interactive workspaces.

2.1 Infrastructures for Interactive Workspaces

Distributed infrastructures such as Gaia [15], iROS [6] and Aura [18] provide systems-level services for application relocation, input redirection, and file sharing in an interactive workspace. Gaia, for example, supports presence detection for users, devices, and services, provides context events and services, and supports an information repository for entities in the workspace [15]. Gaia also provides an application framework to construct or adapt existing applications to execute and be relocated in an interactive workspace [14, 16]. While our toolset was developed for Gaia, it was engineered to be easily portable to other existing infrastructures.

Most modern operating systems enable a single workstation with a multi-head VGA card or multiple VGA cards to provide the ability for a user to seamlessly relocate applications and redirect input among connected screens. By building on top of Gaia, our toolset enables users to redirect input and relocate applications across screens driven by independent, heterogeneous devices that are network connected.

2.2 Interaction Designs for Relocating Applications and Redirecting Input

XWand [20] is a set of wireless sensors packaged in a wand-shaped device that enables a user to control lights, stereos, TVs, and more. VisionWand [4] enables a user to manipulate artifacts on large screens using computer vision to track a passive wand. Although XWand and VisionWand could be extended to relocate applications in an interactive workspace, our interface enables a user to relocate applications not visible to a user, e.g., applications that are on screens turned away from the user, and does not require the user to pick up a separate input device and then switch back.

In [9], researchers extended a browser to enable users to browse web pages across screens driven by independent machines. A user specifies the destination screen from a textual list of available screens. Because our interface provides a visual rather than a textual interface, users can leverage their spatial reasoning abilities to both relocate applications and redirect input.

In I-Land [19] researchers developed several novel interactions to enable a user to relocate applications among screens using gestures. Pick-and-Drop [12] allows users to relocate applications by virtually assigning them to movable physical objects. EasyLiving [3] relocates application windows among screens in a room by passively tracking user movement. Our interface enables a user to relocate an application among screens without being physically close to or physically moving among them.

With UbiTable [17] users can share information on a horizontal work surface using an interface of geometric paths and iconic portals. The shared area of the horizontal surface is used to exchange information among users. In our interface, we enable users to relocate information directly among screens through an iconic, conformal representation of the workspace.

In [13], shared display surfaces are formed by spatially extending a user's local desktop onto the surrounding table surface. To relocate an application, a user hyper drags the application between locations. With our interface, users are able to relocate applications among screens that are not in their field of view. Also, our interface supports both application relocation and input redirection in the same visual interface, allowing users to coalesce these tasks into a single interaction. Finally, our interface is built on top of a toolset that can be ported to other infrastructures, enabling it to be used in a variety of interactive workspaces.

2.3 Toolsets

In Mighty Mouse [2], researchers modified a remote desktop protocol to enable users to redirect input across multiple devices. Users initiate a redirection by selecting a destination screen from a list of identifying icons. To end input redirection, the user performs a special click and key combination. PointRight [8] utilizes the iROS infrastructure to provide configurable geometric paths which enable users to redirect input across devices. In PointRight, users define behaviors to construct the geometric relationship of screens, e.g. moving the cursor off the left side of one screen connects it to the right side of another. In addition to providing support for application relocation, our toolset allows users to visually configure an iconic representation of devices within a physical workspace. This allows users to perform relocation and redirection tasks without having to recall the geometric alignment of screens.

iCrafter [11] automatically generates an interface with a top-down view of the workspace to enable relocation of services. Users access textual lists of services attached to a screen and drag the desired service descriptor to another screen. For example, a user can drag a URL from a laptop to a shared screen to relocate a web browser. In contrast, our work provides effective end user tools that enable iconic representations of various workspaces to be quickly constructed. At runtime, the constructed representation is dynamically updated as events are generated due to changes in the location of applications or devices in the workspace. Additionally, our interface enables a user to visually relocate representations of applications rather than having to mentally map textual identifiers of applications and screens to the corresponding applications and physical screens that they refer to in the workspace.

3 Use and Construction of the Interface

In this section, we describe the iconic representation of our interface, how a user interacts with it to relocate applications and redirect input, and the graphical tool that allows others to rapidly construct iconic representations for their own workspaces.

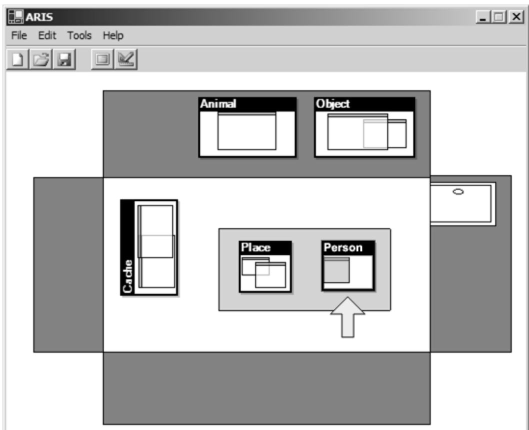

Fig. 1. A screen shot of the iconic representation for our interactive workspace. The representation shows each wall in the workspace pulled down on its back side so that the screens attached to the walls all face upwards. Applications are shown within the screens. The rectangle in the middle of the room represents the table, which has two graphics tablets on it. An arrow shows a user's current location and orientation in the workspace.

3.1 Iconic Representation

As shown in Figure 1, our interface uses an iconic representation of the workspace in a 2-D, foldout view. In a *foldout* view, the walls of the workspace appear pulled down on their back sides so that the screens attached to them all face upwards, providing a complete, distortion-free view of their content. The interface provides representations of all screens connected to the workspace. On each screen, representations of application windows are shown sized and positioned relative to their size and position on the corresponding physical screens. Salient physical objects such as tables, desks and doors are included to enable users to quickly orient the interface's representation to the physical workspace. The representation also contains a yellow arrow indicating a user's current location and orientation in the workspace.

Because the interface provides a spatial representation of the workspace, it supports recognition over recall [10] and enables users to leverage their spatial reasoning abilities to quickly and easily perform relocation and redirection tasks. Our interface is the first to support both of these tasks within a single visual metaphor.

3.2 Relocating Applications and Redirecting Input

A user interacts with the interface's iconic representation to relocate applications and redirect input among screens in the workspace. To relocate an application, a user invokes the interface by selecting a specially added button on the window's title bar, double-clicks its application icon or leaves the interface open indefinitely. The interface receives events from the configuration manager to ensure its representation always reflects the current state of applications, devices, and users in the workspace.

Once invoked, the user selects the representation of the application, drags it to the destination screen in the interface, and drops it. While it is selected, the representation changes color and a rectangular, yellow outline is drawn around the corresponding application in the workspace, which we call a "live outline." As a user is dragging the representation across the screens in the interface, the live outline of the application can be seen moving across the corresponding physical screens in the workspace. If a user looks up from the local screen, the outline provides confirmatory feedback of the ongoing interaction. Once the representation is dropped, the live outline is removed and the interface sends a request to our toolset's Application Manager to relocate and position the application to the destination screen.

Because the iconic representation depicts the *entire* workspace, a user can interact with the interface on a local screen to relocate an application from any screen to any other screen. For example, a user can move an application from one large display to another large display by interacting with the interface executing on their local laptop. Because a relocation task can be completed entirely on a single screen, our interface also supports the use of a stylus input device or touch screen without having to switch it to a relative positioning mode. A relative mode would be required if the cursor had to leave the screen as part of the interaction.

To redirect input using the interface, a user positions the cursor over the destination screen and right-clicks (a stylus and touch panel can also support right-clicks). We chose a right-click interaction for input redirection to disambiguate it from the start of application relocation. The interface sends a request to the runtime engine to redirect mouse and keyboard input to the specified screen. Input can then be redirected back to the local system by performing a similar interaction on the destination screen or by selecting a special key sequence.

3.3 Constructing the Iconic Representation

Since the iconic representation depends on the physical layout of a particular workspace, our toolset offers a graphical tool for rapidly constructing the default representation. In the tool, a user drags representations of walls, displays, doors, and tables from a palette and drops them on the canvas, placing and sizing them as appropriate. To achieve a precise layout, users can enter the exact dimensions of objects in a properties dialog in the construction palette. However, an exact replication is probably not required – the resulting representation just needs to be close enough such that users can quickly associate it with the physical workspace.

Once the default representation is loaded into the runtime system, it can be dynamically updated. For example, when the location or presence of portable devices changes, appropriate events can be sent to the interface to update its representation.

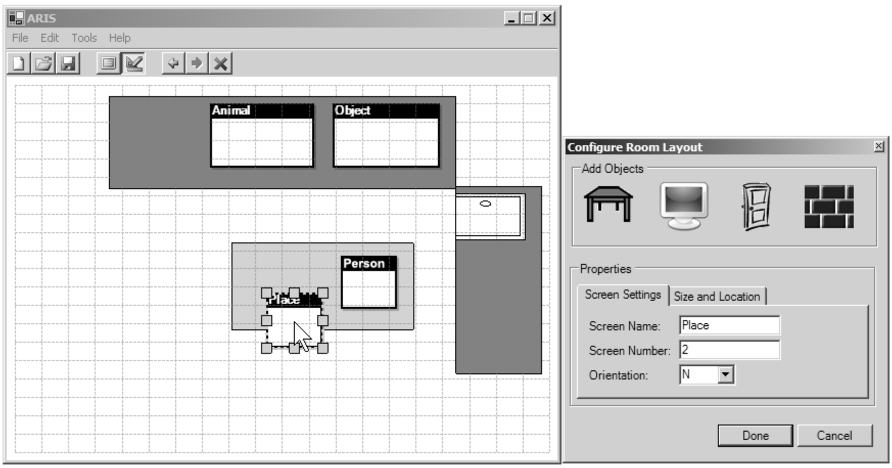

Fig. 2. A screen shot of our graphical construction tool which enables users to quickly construct an initial representation of the workspace. In this screen shot, the user has partially constructed the representation shown in Figure 1.

4 Toolset Architecture and Implementation

In this section we describe the design goals of our toolset, its architecture (shown in Figure 3), and the interface runtime that supports relocation and redirection tasks.

4.1 Design Goals

A design goal of our toolset was to enable it to be used in interactive workspaces driven by different distributed infrastructures. Because there are several such infrastructures for interactive workspaces, e.g. [6, 15, 18], we implemented our toolset for one, Gaia, but engineered the toolset so that it can be easily ported to others. This enables our interface to be used across a variety of existing interactive workspaces.

Another goal was to enable our interface to be used in workspaces with multiple small and large displays driven by a single PC with multi-head graphics cards. While this configuration does not support some of the advanced, real-time features of distributed infrastructures, it does effectively simulate many of the core functions. By supporting this configuration, we offer a low investment mechanism for researchers to use or adapt our interface and compare its use for interactive workspaces. For brevity, the following sections only discuss how our toolset functions with Gaia.

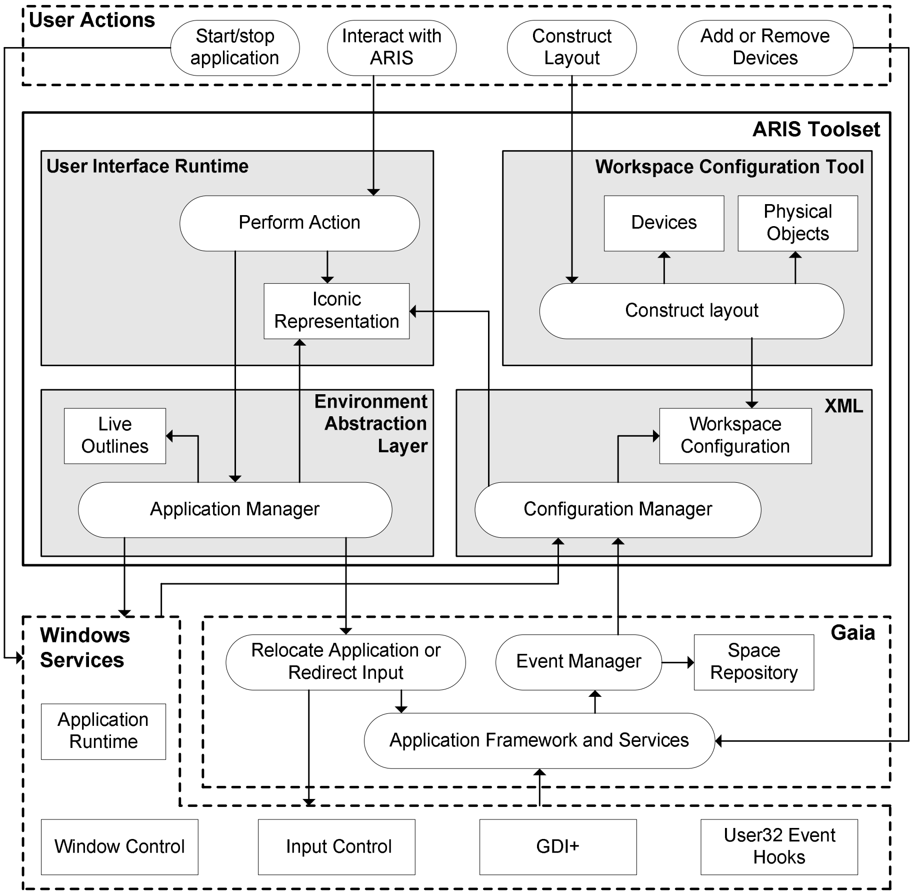

Fig. 3. The architecture of our toolset

4.2 Gaia

Gaia is a distributed infrastructure that manages devices, applications and user state in an interactive workspace. Its application framework [14, 16] provides system support for developing and extending applications which can migrate across heterogeneous devices. Gaia negotiates with the native operating system to acquire application state and coordinate relocation of applications and redirection of input. When users bring devices in or take devices away from the workspace, Gaia's event manager detects the change and notifies our configuration manager (see figure 3). The event manger also notifies the space repository, a service that maintains device, application, and user state for the workspace.

Our toolset extends and abstracts the functionality of Gaia. The toolset's environment abstraction layer (EAL) was created, in part, to provide a simplified relocation and redirection API to the interface runtime. This layer is responsible for translating requests from the interface runtime into a series of requests to the Gaia infrastructure. This intermediate layer enables our interface to execute without direct dependencies on Gaia, making it only necessary to modify the EAL when porting it.

4.3 Environment Abstraction Layer

The Environment Abstraction Layer (EAL) executes on each device in the workspace. In addition to providing a simplified API to the interface runtime, the EAL provides application query services for the interface runtime, coordinates application relocation and input redirection with the underlying infrastructure, and provides support for live outlines. When the interface is invoked, the runtime contacts the EALs executing on each device to acquire information about running applications such as their size, position, and window stacking order.

When the interface runtime sends a request for application relocation or input redirection, the EAL translates the request into the appropriate calls for the underlying infrastructure, which then completes the request. When the EAL executing on the local system receives a completion confirmation from the application framework, it then contacts the EAL on the destination device (which may be the same device) and positions the application window as a function of where it was positioned in the iconic representation. The EAL also coordinates with the configuration manager to update all of the iconic interfaces currently running in the workspace.

To support live outlines for an ongoing relocation task, the EAL is responsible for drawing the rectangular outlines on the appropriate screens. As a user drags a representation of an application across a screen in the interface, the interface runtime contacts the EAL executing on the system driving the corresponding screen to draw or update the outline.

4.4 XML Specification

An XML specification maintains information about the items depicted in the interface's iconic representation. Attributes include object type, location, size, orientation, hostname, screen resolution, etc. The XML specification is created using the graphical tool described in section 3.3.

At runtime, the configuration manager loads the XML specification. When device attributes change, the event manager in Gaia contacts the configuration manger, which updates the specification in memory. The configuration manager notifies the interface runtime, which updates its iconic representation. Through this information loop, the toolset ensures that the iconic representation in each instance of the interface executing in the workspace accurately reflects the current physical state.

4.5 Interface Runtime

The interface runtime draws the iconic representation based on information supplied by the configuration manager. As discussed in section 3.2, a user performs input redirection and application relocation by interacting with the iconic representation. When the user drags an application across screens in the iconic representation, the interface runtime contacts the EAL executing on the system driving the corresponding screen to draw or update the live outline. When a user completes a relocation or redirection interaction, the EAL is contacted to service the request.

5 Usability Evaluation

A user study was conducted to evaluate the effectiveness of the interaction design in our interface and to understand how to improve it. The results also serve another important function – providing an empirical baseline against which the usability of alternative interfaces can be compared. Without the availability of such results, it is difficult to judge the relative quality of alternative interfaces for performing similar tasks. In our study, we used a representative configuration of an interactive workspace and used our graphical tool to construct the corresponding iconic representation.

5.1 Workspace Configuration

Our interactive workspace consisted of three 61” plasma screens 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. We positioned two plasma screens in front of 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. This configuration is representative of existing interactive workspaces, e.g., [7, 14]. For the study, four of the screens were labeled with a category of image content, Person, Place, Animal, or Object, while the fifth screen (one of the three large displays) was labeled Cache.

A high-end Dell Precision 450n workstation was used to drive the screens. 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 for later analysis. We chose to use the single PC configuration of our toolset because the Gaia distributed infrastructure is still a research prototype and we did not want slow response times or other errors in the infrastructure to adversely affect the tasks.

5.2 Users and Task Scenario

Sixteen users (7 female) participated in the study. Users consisted of undergraduate and graduate students, and administrative professionals from our institution. The high-level task scenario was to relocate a PowerPoint application among screens in the workspace and to redirect input to perform annotations. The 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 needed to be relocated to the screen labeled Person), redirected input to the screen closest to them, typed an annotation for the image (e.g., the name of the person in the image), 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 screen labeled Cache.

The task scenario is comprised of representative subtasks commonly performed in interactive workspaces, relocating applications based on their content and redirecting input for annotation. Most importantly, these subtasks were sequenced into a meaningful, higher-level task scenario. We had users perform the tasks in rapid succession to stress the use of the interface.

5.3 Procedure and Measurements

Upon arriving at the lab, we went through an informed consent process with the user. The experimenter described the equipment in the room, explained the task and demonstrated the functionality of the interface. The users perform a practice task consisting of six images; and, if requested, a user could perform a second practice to ensure the interface and task was understood. Once questions were answered, the user performed the experimental task and was instructed to perform the task as quickly and accurately as possible. When finished, the user completed a subjective workload and a post-task questionnaire. In our study, we measured:

- *Time to relocate an application from one screen to another*. Computed from the timestamps in the interaction videos, relocation time was measured from when a user first advanced the slide in the application to when the application appeared on the target screen.
- *Time to redirect input from one screen to another*. Computed using the same method, redirection time was the time to redirect the cursor to the local screen to enter the annotation and then to redirect the cursor back to the screen with the application. The time spent performing the annotation was not included.
- *Errors when relocating an application or redirecting input*. An error was any interaction step that did not move a user closer to completing the task. Example errors would be moving the application to the wrong screen or using a left-click rather than a right-click to perform input redirection.
- *Subjective workload*. This was measured using the NASA TLX [5]. 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 (0) to High (80) for each dimension, and is measured in 1/16" increments.
- *User satisfaction*. Users rated the interface according to ease of use, appropriateness for the task, 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). Users were also asked to briefly explain their ratings.

6 Evaluation Results

In this section we present the results from our usability study.

6.1 Task Performance and Error

Users were able to perform the tasks quickly and with minimal error. For application relocation, users completed the task with a mean performance time of 7.99 seconds (SD=6.96 seconds). Users completed input redirection tasks with a mean performance time of 10.56 seconds (SD=5.08 seconds).

When performing the tasks, the number of errors committed was quite low overall. For application relocation, we identified only a single error out of 48 trials. For input redirection, we identified just three errors out of 48 trials. Most importantly, all users were able to successfully complete the tasks after just a few minutes of instruction on how to use the interface.

6.2 Subjective Workload and User Satisfaction

Figure 4 shows the ratings of subjective workload. The average workload was 25.46 (SD=21.88), or 31.8% of the maximum. The average along each workload dimension was well below the midpoint value, with the highest being mental demand with an average of 46.1%. Overall, the interface induced relatively low workload on a user.

Figure 5 shows the ratings of user satisfaction for ease of use, appropriateness, and ease of learning. On each dimension, users rated the interface highly. On a scale of 1 (worst) to 7 (best), users rated ease of use 5.19 (SD=1.47), appropriateness also 5.19 (SD=1.42), and ease of learning 6.50 (SD=0.82). Results show that users experienced reasonably high satisfaction when performing the tasks with the interface.

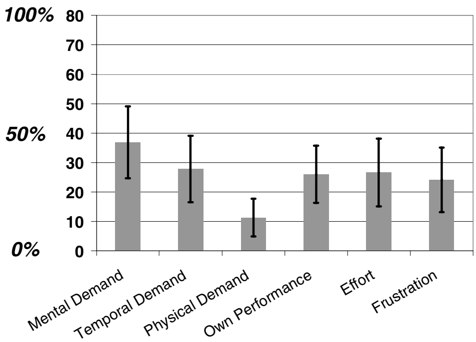

Fig. 4. Subjective workload ratings

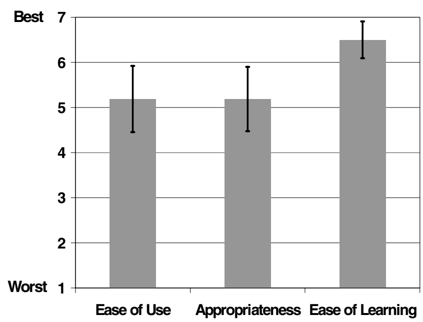

Fig. 5. User satisfaction ratings

7 Discussion and Future Work

The results of our evaluation show that the interaction design of our interface is effective for performing application relocation and input redirection tasks. After just a few minutes of instruction on how to use the interface, users were able to perform relocation and redirection tasks quickly and with minimal error, reported low subjective workload, and experienced high satisfaction. Also, users found the spatial mapping in the iconic representation useful for performing the tasks:

> "Moving of the application is done very easily, the mapping of the room was very accurate and easy to think about when using."

> "[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."

> "[The interface] clearly depicts the whole space on one screen in a manner that is very accessible"

We attribute the low number of errors to the spatial mapping and direct manipulation interaction, but also to the use of the live outlines. We observed that users often looked up into the workspace to see the effects of the ongoing interaction. Observations also showed that users were able to detect and correct errors while in the midst of performing the task. Of the errors that did occur, they were mostly due to the use of the right-click interaction for redirecting input. Users often left-clicked to redirect input and were confused about why no action was invoked. Several users commented that a different interaction technique would be preferred.

Our toolset supports workspaces driven by an existing distributed infrastructure as well as a single PC equipped with multiple graphics cards, supports both application relocation and input redirection tasks in the same visual metaphor, and provides a graphical tool for rapidly constructing iconic interfaces for various workspaces. Our toolset can be downloaded (http://orchid.cs.uiuc.edu) and used for both research and practice in interactive workspaces. Also, the results from our study provide an initial baseline against which alternative interfaces can be compared.

For future work, we plan to:

- *Investigate alternatives to the use of a right click for redirecting input.* A few users left clicked several times before recalling that a right-click was needed. We are exploring alternative interactions such as the use of an 'input redirection' icon that users can drag to the destination screen.
- *Reduce the latency in performing application relocation in a workspace driven by Gaia.* To relocate an application through Gaia, the system takes about two seconds, which causes a noticeable delay and can confuse users. We are working with the Gaia group to reduce this latency.
- *Support more group-based information and interaction in the interface.* While the interface shows application and cursor location information, we want to extend the interface to enable users to manage access permissions for shared displays, to identify specific applications as being "public" and then only show those applications in the interface, and to view which interactions other users are interacting with to better convey activity awareness.

8 Conclusion

To work productively in an interactive workspace, users need effective interfaces to seamlessly share, annotate, and juxtapose information across independent devices. Our research has made several contributions in this direction. We developed an interface that uses an iconic representation of a workspace for performing relocation and redirection tasks, a graphical tool for rapidly constructing iconic representations for different workspaces, and a toolset that provides runtime support for an existing infrastructure and that can be easily ported to others. A usability evaluation showed that the interaction design of the interface is effective for performing relocation and redirection tasks. The interface enabled users to perform tasks quickly and with minimal error as well as to experience low workload and high satisfaction. Results of the evaluation provide the first empirical baseline against which alternative interfaces can be compared. Our toolset can be downloaded today, enabling others to construct and use iconic interfaces for research or practice in their own interactive workspaces.

Acknowledgements

We thank Steve Saville, Chris Drexelius, Carolyn Van Slyck, and Brian Willard for building the graphical construction tool. We also thank Piotr Adamczyk, Shamsi Iqbal, and Damon Cook for offering insightful comments on early drafts of this paper.

References

1. Biehl, J.T. and Bailey, B.P., ARIS: An Interface for Application Relocation in an Interactive Space. *Graphics Interface*, 2004, 107-116.
2. Booth, K.S., Fisher, B.D., Lin, C.J.R. and Argue, R., The "Mighty Mouse" Multi-Screen Collaboration Tool. *UIST,* 2002, 209-212.
3. Brumitt, B., Meyers, B., Krumm, J., Kern, A. and Shafer, S.A., EasyLiving: Technologies for Intelligent Environments. *Handheld and Ubiquitous Computing*, 2000, 12-29.
4. Cao, X. and Balakrishnan, R., VisionWand: Interaction Techniques for Large Displays Using a Passive Wand Tracked in 3D. *UIST*, 2003, 173 - 182.
5. Hart, S.G. and Stateland, L.E. Development of NASA-TLX (Task Load Index): Results of emperical and theoretical research. in Hancock, P.A. and Meshkati, N. eds. *Human Mental Workload*, North-Holland, Amsterdam, 1988, 139-183.
6. Johanson, B. and Fox, A., The Event Heap: A Coordination Infrastructure for Interactive Workspaces. *4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2002.
7. Johanson, B., Fox, A. and Winograd, T. Experience with Ubiquitous Computing Rooms *IEEE Pervasive Computing Magazine*, 1(4): 67-74, 2002.
8. Johanson, B., Hutchins, G., Winograd, T. and Stone, M., PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. *UIST*, 2002, 227-234.
9. Johanson, B., Ponnekanti, S., Sengupta, C. and Fox, A., Multibrowsing: Moving Web Content Across Multiple Displays. *CHI*, 2001, 346-353.
10. 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.
11. Ponnekanti, S.R., Lee, B., Fox, A., Hanrahan, P. and Winograd, T., iCrafter: A Service Framework for Ubiquitous Computing Environments. *UbiComp,* 2001.
12. Rekimoto, J., Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. *UIST*, 1997, 31-39.
13. Rekimoto, J. and Saitoh, M., Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. *CHI*, 1999, 378-385.
14. Román, M. and Campbell, R., Providing Middleware Support for Active Space Applications. *ACM/IFIP/USENIX International Middleware Conference 2003*.
15. Román, M., Hess, C., Cerqueira, R., Ranganat, A., Campbell, R. and Nahrstedt, K. Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing Magazine,* 1(4): 74-83, 2002.
16. Román, M., Ho, H. and Campbell, R., Application Mobility in Active Spaces. *International Conference on Mobile and Ubiquitous Multimedia,* 2002.

17. Shen, C., Everitt, K.M. and Ryall, K., UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. *UbiComp,* 2003.
18. Sousa, J.P. and Garlan, D., Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. *IEEE Conference on Software Architecture*, 2002.
19. Streitz, N.A., et. al., i-LAND: AN Interactive Landscape for Creativity and Innovation. *CHI*, 1999, 120-127.
20. Wilson, A. and Shafer, S.A., XWand: UI for Intelligent Spaces. *CHI*, 2003, 545-552.