An Interface Toolkit for Interactive Workspaces

Jacob T. Biehl and Brian P. Bailey

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL
{jtbiehl, bpbailey}@cs.uiuc.edu

Abstract. To work productively in an interactive workspace, users need effective interfaces to seamlessly share, annotate, and juxtapose digital information across heterogeneous devices. Building on an existing infrastructure for interactive workspaces, we developed an interaction design that uses an iconic representation of a workspace to enable application relocation and input redirection tasks to be performed, a graphical tool for constructing the iconic representations, and a toolkit that provides runtime support for the interface. The iconic representation supports recognition over recall and the use of spatial memory when performing relocation and redirection tasks. A usability evaluation showed that the interaction design of our interface enables tasks to be performed quickly and with minimal error, induces low workload, and supports high satisfaction. While the toolkit was implemented on top of an existing infrastructure, it can be easily ported to others. Our toolkit can be downloaded today, and can facilitate more productive use of interactive workspaces for individual and group work.

1 Introduction

An interactive workspace is a physical workspace that connects the use of small and portable devices, and large shared displays through a distributed software infrastructure. Interactive workspaces can dramatically improve how users share, annotate, and juxtapose digital information for individual and group work [6].

While services for application relocation, input redirection, and file sharing are supported within distributed infrastructures for interactive workspaces, users need effective interfaces for quickly and easily performing application relocation, input redirection, and other tasks. To support these tasks, several interfaces and supporting toolkits have been developed [5, 15, 18]. However, these systems are either too heavily tied to a specific infrastructure, making it difficult to port them to other infrastructures, the effectiveness of the interaction designs in the interfaces have not been evaluated, or variations of the interfaces are overly difficult to construct for different workspaces.

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

discuss the interaction design of the interface, describe an easy-to-use graphical tool for rapidly constructing iconic representations for different workspaces, describe the architecture of our toolkit that provides runtime support for the interface, and present results from a usability evaluation of the interaction design in our interface. The usability evaluation showed that the interaction design in our interface enables tasks to be performed quickly and with minimal error, induces low subjective workload, and supports high satisfaction across users.

While the toolkit was implemented on top of an existing distributed infrastructure, it was engineered to make it easily portable to others, thus enabling our interface to be more widely available. Our toolkit can facilitate more productive use of interactive workspaces for individual and group work and is available for download today.

2 Related Work

In this section, we discuss distributed infrastructures for interactive workspaces, interactions for performing application relocation and input redirection tasks, and toolkits for creating and running interfaces in interactive workspaces.

2.1 Infrastructures for Interactive Workspaces

Distributed infrastructures such as Gaia [15], iROS [5] 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]. Most importantly, Gaia provides an application framework to construct or adapt existing applications to execute and be relocated in an interactive workspace [14, 16]. Our toolkit builds on a specific infrastructure, Gaia, to provide an effective user interface for performing application relocation, input redirection, and other tasks in an interactive workspace. However, the toolkit was also engineered to be portable to other 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 toolkit enables users to redirect input and relocate applications across screens driven by network connected, heterogeneous devices.

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 [8], researchers extended a browser to enable a user to browse web pages across multiple screens. A user specifies the destination screen from a textual list of available screens. Because our interface provides a visual rather than a textual method of relocating applications, users can utilize their spatial memory to perform relocation tasks. Also, our interface supports input redirection and enables a user to manage many multiple applications.

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. A usability evaluation also shows that our interface is effective for performing relocation and redirection tasks.

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 representation of the workspace. We also provide a graphical tool for rapidly constructing iconic representations of other workspaces in which our interface is to be used.

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. Through the iconic representation in our interface, users can relocate applications among screens not in their field of view, and it can be extended to support other activity information. Also, a usability evaluation showed that the use of the iconic representation was effective for performing relocation and redirection tasks. Our interface is also built on top of a toolkit that can be ported to other infrastructures, enabling it to be more widely used.

2.3 Toolkits

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 [7] utilizes the iROS infrastructure to provide configurable geometric paths which enable users to redirect input across devices in an interactive workspace by seamlessly allowing them to move the cursor 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 toolkit 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 in a workspace.

With Pebbles [10] multiple users can use multiple PDAs to simultaneously redirect input to a shared application on a PC. Although this concept could be extended to support multiple shared screens, our toolkit enables users to redirect input and relocate applications using *any* connected device in the workspace.

iCrafter [11] automatically generates an interface that enables a user to interact with and relocate services in an interactive space. After selecting services from a textual list, part of the generated interface includes a top-down view of the space that enables a user to drag data from one screen and drop it on another, e.g., a user could drag a URL from a laptop to a shared screen to change the location of a web browser.

In contrast to iCrafter, our toolkit allows the *user* to construct an iconic representation of a workspace that is spatially correct and updates as the location of applications and devices change. Additionally, our interface enables a user to visually relocate representations of application windows rather than having to mentally map textual names of applications and screens to the actual applications and physical screens that they refer to in the workspace.

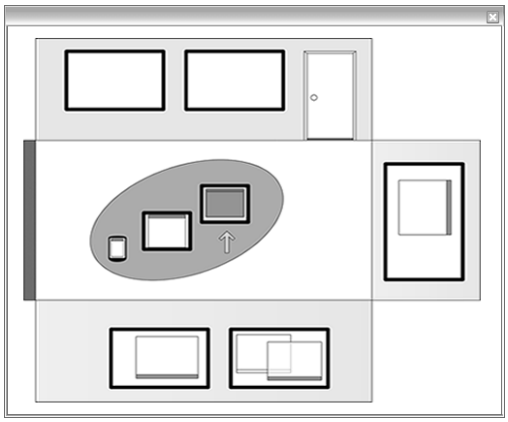


Figure 1: A screen shot of our interface which shows the iconic representation of an interactive workspace. The representation shows each wall in the workspace pulled down on its back side so that the screens attached to the walls face upwards. Applications are show within the screens. The oval shape represents the table which has two graphics tablets and a PDA on it. A yellow arrow shows a user's current location and orientation in the workspace.

3 Use and Construction of the Interface

In this section, we describe the iconic representation of our interface, a graphical tool for constructing the representation, and how a user interacts with the representation to relocate applications and redirect input in an interactive 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. Within the foldout view, the interface provides representations of all screens - large displays, smaller and portable devices - connected to the infrastructure driving the workspace.

Within each screen's representation, the interface shows iconic representations of application windows sized and positioned relative to their size and position on corresponding physical screens. Salient physical objects such as tables, desks and doors may also be included to enable users to quickly orient the representation to the workspace. A yellow arrow shows a user's current location and orientation in the workspace.

Because the interface provides a visual, spatial representation of the workspace, it supports recognition over recall [9] and enables users to leverage their spatial memory 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.

The iconic representation is specified in an XML file. At startup, the configuration manager of the toolkit loads the representation in the file. However, the representation can be updated at runtime through a network connection to the configuration manager. This enables a workspace to notify the interface of dynamic changes such as devices being brought into or taken away from the workspace.

```
<?xml version="1.0"?>
<Settings>
  <Container>
    <Width>200</Width>
    <Height>200</Height>
  </Container>
  <Doors>
    <Rectangle>
      <Location>
        <X>10</X>
        <Y>60</Y>
      </Location>
      <Size>
        <Width>35</Width>
        <Height>25</Height>
      </Size>
      <X>10</X>
      <Y>60</Y>
      <Width>35</Width>
      <Height>25</Height>
    </Rectangle>
  </Doors>
  <RoundTables>
    <RoundTableObject>
      <Tilt>0</Tilt>
      <RectObject>
        <Location>
          <X>85</X>
          <Y>72</Y>
        </Location>
        <Size>
          <Width>165</Width>
          <Height>100</Height>
        </Size>
        <X>85</X>
        <Y>72</Y>
        <Width>165</Width>
        <Height>100</Height>
      </RectObject>
    </RoundTableObject>
  </RoundTables>
  <Screens>
    <ScreenObject>
      <HostName>cs-chamomile</HostName>
      <Display>1</Display>
      <Orientation>N</Orientation>
      <Resolution>
        <Width>1280</Width>
        <Height>1024</Height>
      </Resolution>
      <RectObject>
        <Location>
          <X>100</X>
          <Y>92</Y>
        </Location>
        <Size>
          <Width>22</Width>
          <Height>20</Height>
        </Size>
        <X>100</X>
        <Y>92</Y>
        <Width>22</Width>
        <Height>20</Height>
      </RectObject>
```

Figure 2: An excerpt of the XML file that specifies the iconic representation.

3.2 Constructing the Iconic Representation

Because an iconic representation depends on the configuration of the corresponding workspace, our toolkit provides a graphical tool for rapidly constructing the representation. The configuration tool provides a fold-out view of an empty workspace as a starting point, which can be changed to accurately depict the dimensions of the physical workspace. A user can then drag representations of devices and physical objects such as doors and tables to the canvas, placing and sizing them as needed to depict the workspace. The tool exports the configuration to the XML specification loaded by the configuration manager at runtime. Figure 2 shows the XML specification exported from the tool.

3.3 Relocating Applications and Redirecting Input

At runtime, a user interacts with the iconic representation to relocate applications and redirect input among screens in the workspace. To relocate an application, as shown in Figure 3, a user invokes the interface by selecting a specially added button on the window's title bar. Our toolkit hooks into the Windows event stream to automatically place this button on the title bar of all executing applications. Alternatively, a user can invoke the interface by double-clicking on its application icon or can leave the interface open indefinitely. If left open, the interface receives periodic updates from the configuration manager to ensure that its representation reflects changes in the workspace.

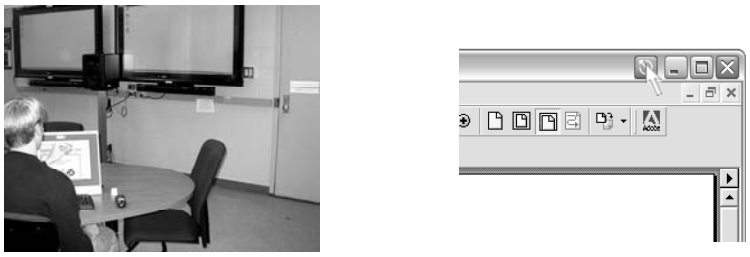
Once invoked, the user selects the representation of the application, drags it to the destination screen in the interface, and drops it. While it is selected, the representation changes color and a rectangular, yellow outline is drawn around the corresponding application in the workspace, which we call a "live outline." As a user is dragging the representation across the screen icons 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 the runtime engine of the toolkit to relocate the application to the destination screen and position it as specified.

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 on their laptop. Because a relocation task can be completed on a local screen, our interface also supports the use of a stylus input device without having to switch it to a relative positioning mode. A relative mode would be required, for example, if the cursor had to leave the local screen as part of the interaction.

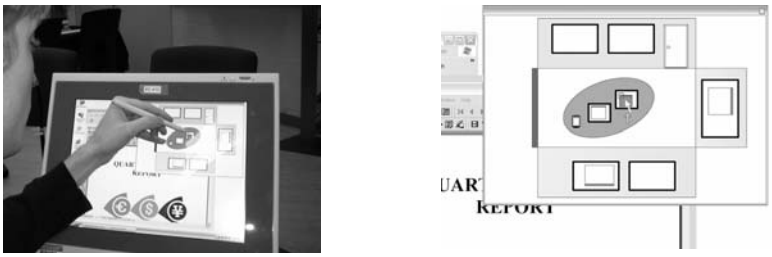
To redirect input using the interface, a user positions the cursor over the destination screen and right-clicks (a stylus and touch panel can also support right-clicks). The interface sends a request to the runtime engine to redirect mouse and keyboard input to the specified screen. Input can then be redirected back to the local system by performing a similar interaction with the interface on the destination screen or by

selecting a special key sequence. We chose a right-click interaction for input redirection to disambiguate it from the start of application relocation.

Global View **Close-up View**



(a) User selects the title bar invocation button to bring up the interface.



(b) The iconic representation appears and the user begins to drag an application's representation.

(c) The user drags the application to the desired destination screen. Notice how an outline of the application's current location is shown on the physical screen.

(d) The user releases from the drag operation which completes the relocation action.

Figure 3: Using the interface to relocate an application. The left shows a global view of the workspace while the right shows a close-up of the interaction on the local screen.

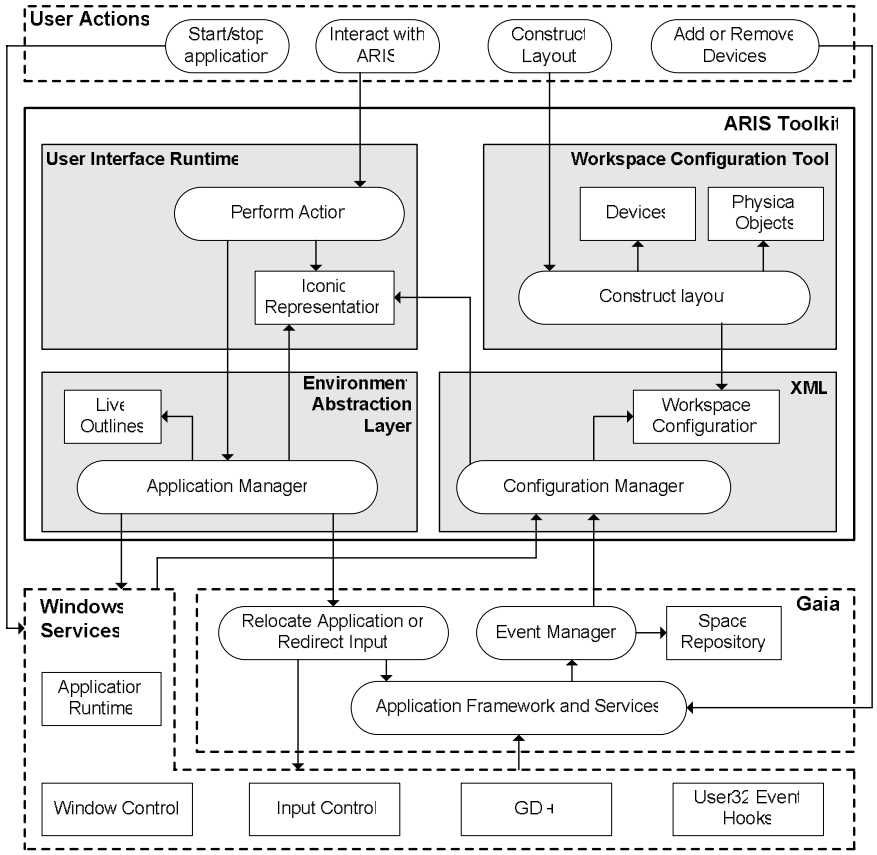


Figure 4: The architecture of our toolkit.

4 Toolkit Architecture and Implementation

In this section we describe the goals of our toolkit, its architecture, and how it supports relocation and redirection tasks. Figure 4 shows the toolkit architecture.

4.1 Design Goals

A goal of our toolkit was to enable our interface to be used in interactive workspaces driven by different distributed infrastructures. Because there are several existing infrastructures for interactive workspaces [5, 15, 18], we implemented our toolkit for one such infrastructure, Gaia, but engineered our toolkit so that it can be easily ported to others. This would enable our interface to be more widely used.

Another goal was to enable our interface to be used in interactive workspaces driven by a single PC with multi-head graphics cards. While this configuration does not support many of the advanced, real-time features of distributed infrastructures, it does enable multiple small and large screens to form a connected workspace. Rapid advances in graphics hardware have also made this configuration cost effective and simple to setup. For brevity, however, the following sections only discuss how our toolkit was implemented on top of Gaia.

4.2 Gaia

Our toolkit was built on top of Gaia, a distributed infrastructure that manages devices, applications and user state in an interactive workspace. Gaia's application framework [14, 16] can be used to develop or extend applications so they 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 4). The event manger also notifies the space repository, a service that maintains device, application, and user state for the workspace.

When applications are started or stopped in the workspace, the native OS informs Gaia of the changes, which then updates the space repository. When an application relocation or input redirection request is received, Gaia's application framework coordinates the relocation or redirection task with the native OS.

Our toolkit extends and abstracts the functionality of Gaia. The environment abstraction layer enables the interface to execute without direct dependencies on the supporting Gaia infrastructure, making it easy to port to other infrastructures. We describe this layer of abstraction next.

4.3 Environment Abstraction Layer

The Environment Abstraction Layer (EAL) executes on each device in the workspace. We constructed the EAL as part of our toolkit to provide application query services for the interface runtime, coordinate application relocation and input redirection with the underlying infrastructure, and provide support for live outlines. When the interface is invoked, it contacts the EAL that is executing on each device to acquire information about running applications such as their size, position, and 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 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 based on where it was positioned in the iconic representation in the interface. The EAL coordinates with the configuration manager to update each iconic representation in the interfaces executing 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 in the workspace. 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

The XML specification maintains information about devices, physical objects and room dimensions. The XML specification can be created using the graphical tool described in section 3.2. As shown in figure 2, the specification contains properties for the objects in the workspace. These properties include type, location, orientation and size information. For a device object, the specification also maintains properties for the device's hostname, screen resolution, and the window display order.

At runtime, the configuration manager loads the XML specification. When changes to devices occur, the event manager in Gaia contacts the configuration manger, which updates its representation in memory. The configuration manager notifies the interface runtime, which updates its iconic representation. Through this information loop, the toolkit ensures that the iconic representation in each interface executing in the workspace always reflects the current state.

4.5 Interface Runtime

The interface runtime draws the iconic representation based on information supplied by the configuration manager. As discussed in section 3.3, 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.

4.6 Implementation Technologies

We developed our toolkit using Microsoft Visual C# .Net and Visual C++ to create COM components that integrate with the application framework in Gaia. We draw the iconic representation using GDI+. By hooking into the Windows event stream, our toolkit is able to add the button for invoking the interface on the title bar of each executing application. The EAL is implemented using SOAP to enable information to be exchanged as serialized objects.

5 Usability Evaluation

We conducted a user study to evaluate the usability of the interaction design in the interfaces that can be constructed with our graphical tool. The evaluation tested how well users could use the interface for a particular workspace configuration. In the study, users performed application relocation and input redirection tasks with the interface. We use the results to understand how to improve the interaction design of the interface. Results also provide an empirical baseline for comparing future refinements to our interface as well as alternative interaction designs.

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, and had resolution of 1280x1024. We positioned two plasma screens behind a 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. This configuration is representative of existing interactive workspaces, e.g., [6, 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 toolkit because the Gaia distributed infrastructure is still a research prototype and we did not want slow response times or other errors in the underlying system to adversely affect users.

5.2 Users and Task

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.

The task 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., 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 screen labeled Cache. This task was representative of those commonly performed in an interactive workspace since a user had to relocate an application

based on its content and redirect input for annotation. 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 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 subjective workload and a post-task questionnaire. The study lasted about twenty minutes.

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. We did not include the time users spent performing the annotation. Measurements were also computed from 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 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 [c6]. 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. This was measured using a ruler in 1/16” increments starting from the Low mark.
- *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 why they gave these ratings.

6 Evaluation Results

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

6.1 Task Performance and Error

All users successfully completed the tasks with minimal instruction on how to use the interface. In addition, 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 and a standard deviation of 6.96 seconds.

Users completed input redirection tasks with a mean performance time of 10.56 seconds with a standard deviation of 5.08 seconds. When performing the tasks, the number of errors committed by users 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.

6.2 Subjective Workload

Figure 5 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

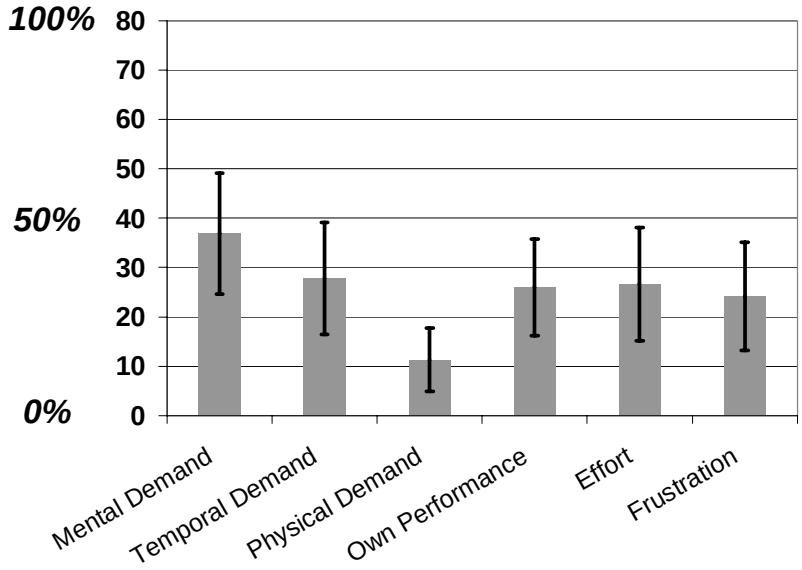


Figure 5: Subjective workload average and error for each dimension.

and these values provide an important empirical baseline for comparison with future refinements to our interface or alternative interfaces.

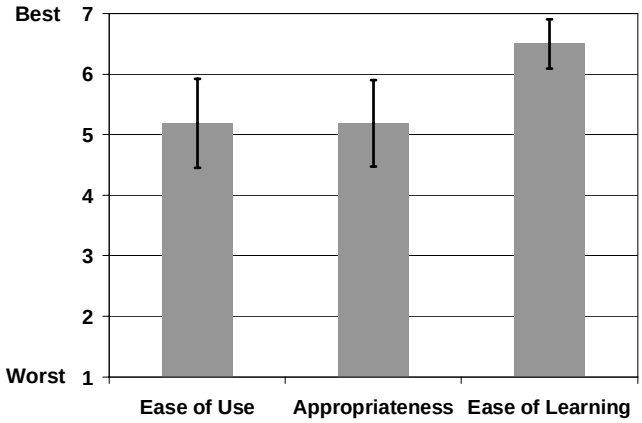


Figure 6: User satisfaction ratings.

6.3 User Satisfaction

Figure 6 shows the ratings of user satisfaction for ease of use, appropriateness, and ease of learning. On each dimensions, 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 easy of learning 6.50 (SD=0.82). Results show that users experienced high satisfaction when performing the tasks with the interface.

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 interact with the interface, users were able to perform tasks quickly and with minimal error, reported low subjective workload, and had high satisfaction. Users found the spatial mapping in the iconic representation useful for performing the tasks. User comments included:

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

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

> "Its like the physical environment I am sitting in. So its easier to correlate to the real environment and start where I left off."
>
> "The mapping was very accurate and easy to think about when using."

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.

For the evaluation discussed in this paper and for other research, we have used our graphical configuration tool to construct iconic representations of multiple workspace configurations. In each case, we were able to use the tool to successfully construct and use the representation without having to manually modify the underlying XML specification.

Our toolkit 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 interface, and provides a graphical tool for rapidly constructing the interface. An evaluation showed that the interaction design in the interface is effective for performing application relocation and input redirection tasks in an interactive workspace. Our toolkit can be downloaded today (location removed for blind review), and can facilitate more productive use of interactive workspaces for individual and group work.

For future work, we plan to:

- *Investigate alternatives to the use of a right click for redirecting input.* To disambiguate input redirection from the start of application relocation (left click, then drag), the interface uses 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 today, we want to enhance our toolkit 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 in the interface, and to view which applications 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 heterogeneous devices. Our research has made several contributions in this direction. Building on an existing infrastructure for interactive workspaces, we developed an interaction design that uses an iconic representation of a workspace for performing application relocation and input redirection tasks, a graphical tool for rapidly constructing iconic representations for different workspaces, and a toolkit that provides runtime support for the interface. The iconic representation supports recognition over recall and the use of spatial memory when performing relocation and redirection tasks. A usability evaluation showed that the interaction design of the interface enables these tasks to be performed quickly and with minimal error, induces low workload, and supports high satisfaction among users. Results of the evaluation provide an important empirical baseline for comparing refinements to our interface and alternative interfaces in the future. While our toolkit was implemented on top of an existing infrastructure, it was engineered to be easily portable to others. Our toolkit can be downloaded today, and can facilitate more productive use of interactive workspaces for individual and group work.

9 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. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2002, 209-212.
3. Brumitt, B., Meyers, B., Krumm, J., Kern, A. and Shafer, S.A. EasyLiving: Technologies for Intelligent Environments. *Proc. 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. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2003, 173 - 182.
5. Johanson, B. and Fox, A. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2002.
6. Johanson, B., Fox, A. and Winograd, T. Experience with Ubiquitous Computing Rooms *IEEE Pervasive Computing*, 2002, 67-74.
7. Johanson, B., Hutchins, G., Winograd, T. and Stone, M. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2002, 227-234.
8. Johanson, B., Ponnekanti, S., Sengupta, C. and Fox, A. Multibrowsing: Moving Web Content Across Multiple Displays. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2001, 346-353.
9. 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.
10. Meyer, B.A., Stiel, H. and Gargiulo, R. Collaboration Using Multiple PDAs Connected to a PC. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 1998, 285-294.

11. Ponnekanti, S.R., Lee, B., Fox, A., Hanrahan, P. and Winograd, T. iCrafter: A Service Framework for Ubiquitous Computing Environments. *Conference on Ubiquitous Computing*, 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*, 2002, 74-83.
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. *Proceedings 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. and al., e. 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.