

Issues in Personalizing Shared Ubiquitous Devices

Jonathan Trevor, David M. Hilbert, and Bill N. Schilit

Fuji-Xerox Palo Alto Laboratory, 3400 Hillview Avenue, Palo Alto, CA 94304 USA
{trevor,hilbert,schilit}@fxpal.com

Abstract. As ubiquitous computing becomes widespread, we are increasingly coming into contact with “shared” computer-enhanced devices, such as cars, televisions, and photocopiers. Our interest is in identifying general issues in *personalizing* such shared everyday devices. Our approach is to compare alternative personalization methods by deploying and using alternative personalization interfaces (portable and embedded) for three shared devices in our workplace (a presentation PC, a plasma display for brainstorming, and a multi-function copier). This paper presents the comparative prototyping methodology we employed, the experimental system we deployed, observations and feedback from use, and resulting issues in designing personalized shared ubiquitous devices.

1. Introduction

Everywhere we go, “shared” devices surround us: TVs, stereos, and kitchen appliances in the home; copiers, fax machines, and projectors in the workplace; ATMs and Coke machines in public. Because these devices don’t know who we are, they provide the same interface, information, and functionality to everyone. This lack of personalization in the real world is reminiscent of the World Wide Web in its infancy—no matter who you were, you saw the same Web pages as everyone else. Today, Web sites remember who you are and tailor their content to match your needs. Amazon.com leverages personalization not only to expedite your purchases, but also to suggest things you may wish to buy based on past purchases, and purchases of others with similar interests. MyYahoo.com dynamically creates pages “just for you” including: your daily horoscope; local weather; and news stories, sports scores, and stock quotes matching your interests. Thanks to personalization, the Web is much friendlier, more efficient, and profitable today. Our research is in applying Web-style personalization to shared real world devices, such as projectors, shared displays, and copiers. Our hope is that this will make real world devices friendlier, more efficient, and more capable than before.

The contributions of this paper are threefold: we promote an experimental methodology, *comparative prototypes*, for evaluating ubiquitous systems; we describe the design and implementation of a novel system for personalizing multi-user document devices in the workplace; and we highlight lessons learned from our experience to inform future efforts in personalizing shared ubiquitous devices.

1.1. Background

Personalization, or end-user customization and tailoring, has a rich history in software applications and environments. Emacs [15] is well known for being end-user customized for a wide range of tasks, from program editing to file browsing and news reading. Many interactive applications support end-user tailoring of toolbars and menus. Beyond personalizing individual applications, people have developed techniques for “teleporting” one’s personal computing environment from one device to another as a way of personalizing computing devices other than one’s own desktop PC [12][8]. The research literature in human-computer interaction and intelligent user interfaces includes many examples of personalizing single-user applications, and the literature in computer-supported collaborative work, operating systems, and the Web covers a range of approaches for personalizing multi-user computing environments. Our interest, in contrast, is in applying this well-known concept to *shared everyday devices*, such as photocopiers.

We are not alone in wanting to personalize the real world. The iCrafter iRoom [11], for instance, presents users with preferred user interfaces for interacting with different classes of devices in the environment, such as light switches. Some researchers have developed physical bookmarks that people can use to mark their positions in TV shows in one room so they can “personalize” TVs in other rooms to resume their viewing [17]. Indeed this trend is moving beyond the research lab and into commercial products. The BMW 7 Series automobile remembers drivers’ seat, mirror and stereo settings and recalls them automatically when drivers use their unique ignition keys to enter the car. Our work is situated in this tradition of personalizing shared everyday devices.

2. Designing and Evaluating a Personal Ubiquitous System

Our research seeks to understand how to leverage personalization to improve our interactions with shared ubiquitous devices. A common approach in ubiquitous computing research is to build and deploy working systems to observe them in use. This has produced numerous insights regarding systems and their affects on users and social contexts. However in deploying “real world” systems, we must choose between a myriad of design alternatives, some of which have far-reaching and unanticipated implications. Once we have designed and deployed a system, how can we know what “might have been” had we chosen different alternatives?

We were particularly concerned with deciding how to best deliver personalization to shared devices: (1) we could integrate personalization with an already existing (or co-located) user interface for controlling the device: an “embedded” approach; or (2) we could provide a personal interface that users access through their own portable devices, such as mobile phones or PDAs: a “portable” approach. Rather than picking one approach and simply “validating” it, we wanted to perform a *comparative evaluation* to gain more general insights to inform future designs. Thus, we were faced with the challenge of evaluating a “real world” ubiquitous system.

2.1. A Comparative Prototype Research Methodology

Evaluating ubiquitous systems is hard, and has attracted the attention of others in the research community [13]. These investigators, like others in CSCW [5][18], argue there is a basic mismatch between traditional evaluation techniques and the needs posed by ubiquitous systems. Namely, these systems are embedded in a variety of complex real world environments that cannot be easily modeled (as required by theoretical analyses), simulated, measured, or controlled (as required by laboratory experiments). These concerns are shared by Abowd, Mynatt and Rodden, who argue “deeper evaluation results cannot be obtained through controlled studies in traditional, contained, usability laboratory.” [3]. As a result, many investigators have abandoned traditional comparative evaluation techniques and opted instead for techniques adapted from the social sciences, such as anthropology.

We wanted to perform a comparative evaluation similar to a laboratory experiment, but in such a way that we could observe the effects of our design decisions in relatively unconstrained, real world use. This led us to the following process:

1. **Design with alternatives.** Pick a design decision to vary and build prototypes (or configurations of an existing system) to embody multiple design alternatives.
 - a. Make the system variants provide the same basic underlying functionality.
 - b. Make the underlying system functionality as simple as possible but at the same time compelling and useful enough to produce real usage.
2. **Vary deployment situations.** Deploy the prototypes in varied situations to help answer the question: are observed effects linked to a single situation or are they more general?
3. **Compare and contrast.** Use qualitative or quantitative data collection and analysis techniques to compare and contrast the alternatives and situations.
 - a. Try to determine whether observed effects vary depending on the design alternatives, the varied situations, or both.

In addition, the prototypes (or configurations) need to be useful—they need to be used by real people to address real problems and fit in with existing practices. This should be accomplished through techniques such as informal observation of current work practices and problems, or by using more principled studies.

This evaluative method gives us a framework for better understanding a ubiquitous computing system. It goes beyond *designing for use* and promotes *designing for evaluation*. The framework produces a matrix of observations and measurements across situations of use that help us to think about ubiquitous computing design. At the end of the paper we present a summary of findings with this matrix.

The next subsections discuss how we applied this methodology. First we introduce the PIPs system. Then we discuss the system variants embodying our *design alternatives*. Finally, we discuss the locations, devices, and tasks that made up our *varied situations*. Design and implementation details are presented in Section 3. Observations and resulting issues are presented in Section 4.

2.2. Personal Interaction Points (PIPs) System

We set out to design a personal interaction points (PIPs) system for personalizing shared pervasive devices. We were motivated, in part, by the copiers, fax machines, and printers that are so common in hotels business centers, convenience stores, and copy shops. Although there are many personalization features imaginable, we decided that integrating personal computer file access with a shared device's functions would be a good starting point. The PIPs system embodies this by giving "smart" access to people's file reference history *at* shared devices, just like the Windows recent "Documents" menu gives access at desktop computers. The "smart" part of the system was to match file types from the user's "information cloud" with the function types of devices. For example, if the device is a projector in the conference room then the preferred file type is PowerPoint.

At the same time as we developed the underlying system functionality of providing a file history at shared devices, we began to consider how best to evaluate the resulting system. Although we wanted to add a useful personalization system to our workplace, we were also willing to learn from and then throw away the first system, so we could build a better second system. This led us to focus on better understanding the design alternatives as well as the deployment situations. These are described below.

2.3. Design Alternatives: Embedded versus Portable Interfaces

Many ubiquitous systems, e.g., PalPlates [7], adopt an *embedded* model in which users interact directly with devices embedded in the environment. Other ubiquitous systems, such as CybreGuide[1], adopt a *portable* approach in which users interact with portable devices, such as their cell phones or PDAs. In the PIPs system, we wanted to vary this design decision to learn more about the relative merits of the design alternatives.

Thus, we developed two PIPs variants: embedded PIPs which provide an interface connected to the device itself, e.g., a touch screen (figure 1 left); and portable PIPs which users access via portable devices, e.g. mobile phones or PDAs (figure 1 right). We varied this decision because we were not certain which approach would be most appropriate. For instance, would users feel comfortable accessing their private data via a public device? Would they feel better accessing their data via their own cell phone? What if users forget their cell phones or wireless connectivity is unreliable? Are larger, embedded user interfaces inherently more usable than tiny, portable interfaces provided by cell phones and PDAs?

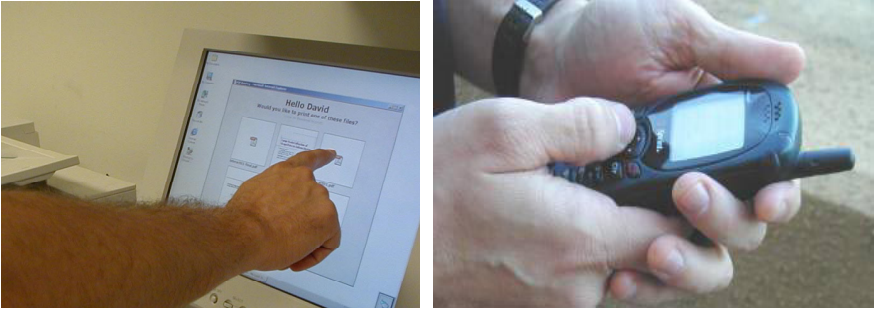


Fig. 1. Two key design alternatives for ubiquitous systems: embedded (left) or portable (right) interfaces.

The PIPs system supports both design alternatives by providing two types of user interfaces: one available at the device and accessed through an embedded touch-screen, and the other which is accessed via a wireless laptop or PDA.

2.4. Varied Situations: Presentations, Brainstorming, and Printing

In order to understand the effects of multiple situations, we took three shared devices that support different tasks in different locations in our workplace, and integrated the PIP system into each. We selected the presentation PC in our formal conference room, a large plasma display in our brainstorming room, and a multi-function copier in our mailroom. The PIP in each location helps users do the same tasks they would normally do in that location, but via a personalized interface that streamlines common activities.



Fig. 2. Three places with shared devices in our laboratory: The formal conference room, the informal brainstorming room, and the mailroom.

In the following subsections we describe each situation, including: location, shared device, and common tasks; observations regarding how personalization might help; and a brief description of the PIP variant deployed in each case.

2.4.1. The Podium PC in Our Formal Conference Room

The speakers' podium in our formal conference room (Figure 2 left) is equipped with a shared PC that drives two screens: a small monitor in the podium itself and the room's main projection display. The podium also provides a console for controlling the room's AV setup, e.g. the room lights, volume, and blanking of the main screen. Presenters take turns using the podium PC to show their presentations on the main projection screen.

Observations of our colleagues showed considerable time, sometimes several minutes, was spent searching for presentations stored on the network. This resulted in awkward moments while presenters searched for their files, significantly breaking up meeting flow, particularly in meetings with multiple presenters. To avoid this, many users adopted the practice of copying their presentations to a shared folder on the podium PC ahead of time. However, this requires pre-planning and effort as well as manual re-synchronization of files whenever users update their presentations either at their office PCs or at the podium PC.

Thus, we saw the opportunity to personalize the activity of locating presentations and displaying them on the main projection screen. Our "presentation PIP" provides users with a personal interface for accessing their most recently edited or viewed presentation, which they may then display on the main screen with a single interaction. No file copying or synchronization is required as documents are securely accessed over the network from their original locations.

2.4.2. The Plasma Display in Our Brainstorming Room

Our brainstorming room (Figure 2 center) is a much smaller room intended for use by about eight or ten people for informal discussions and brainstorming. It contains a 42-inch plasma display that people can use to present and edit documents via laptops. In contrast to our formal conference room, documents accessed in the brainstorming room are typically "works in progress", such as unfinished papers and presentations, or electronic summaries of discussions and brainstorming.

Observations of our colleagues indicated that the plasma display was not being used often. This was because most people use their own office PCs to do their work, and there are only a few laptops for people to share. Thus, using the plasma display required pre-planning and effort to move one's documents to a shared laptop (if one was available).

It seemed that personalization could streamline the activity of accessing and editing working documents on the brainstorming display. Rather than relying on laptops to connect to the display, we provided a networked PC running the PIP software, and added a touch screen overlay and wireless keyboard to expedite interactions at the display. Our "brainstorming PIP" provides users with a personal interface for quickly accessing their most recently edited or viewed documents, which they may then view and edit on the plasma display. Users may also create new documents, e.g., for note-taking, which are then automatically saved back to their office PCs.

2.4.3. The MFD in Our Mailroom

Finally, our mailroom (Figure 2 right) hosts a multi-function copier device (MFD) with print, copy, scan, and fax functions. The MFD is networked and centrally located within the building and used by nearly everyone daily.

Observations of our colleagues showed people most frequently used the MFD to print documents from their offices. While the MFD serves this purpose well, we noticed some areas for improvement that might be addressed by a personalized interface. First, after initiating print jobs, users must walk from their offices to the mailroom to pick them up. If the job didn't print the way they wanted, they must walk back to their office, re-initiate the job, and then walk back to the mailroom again. Users who wanted to print sensitive documents (not knowing how to use the MFD's print-and-hold function) sometimes started jobs from their offices, and then ran to the mailroom to pick them up before others could see them.

Thus, we saw an opportunity for reducing the need for users to run back and forth between the mailroom and their offices. We added a 15-inch touch screen monitor co-located with the MFD, which we connected to a PC running the PIP software hidden in a cabinet beside the MFD. Our "printer PIP" provides users with a personal interface for accessing their most recently edited or viewed documents, which they may then print on the MFD with a single interaction.¹

3. Design and Implementation

Broadly speaking, there are three key requirements any system for personalizing shared devices must address: (1) identifying users, (2) learning and remembering things about users, and (3) creating a personalized interface for operating the device. In addressing these requirements, we consciously tried to support and integrate the day-to-day work practices of our laboratory to increase the likelihood our system would actually be used.

Personalized systems must **identify users** before they can provide personalization. For instance, ATMs use cards with magnetic strips plus PINs to identify users. In our laboratory, everybody already has a username and password that they regularly use to log into their office PC. The PIP method of identifying the user builds on top of this existing practice and uses the same Microsoft Windows NT security model for identifying and authenticating users. For our embedded PIPs, we exploit RFID technology [16], a type of contactless smart card, to enable users to quickly and easily identify themselves to the embedded interfaces without having to type their usernames and passwords. Users need only carry a small "tag" with them, which can be detected when waved near special tag readers co-located with the shared devices.² If a user forgets their tag, the embedded interfaces allow them to type in their NT username and password manually. In the case of the portable PIPs, users log in using this second method, and then identify which device they want to interact with by selecting from a list of available devices.

¹ Although we have not yet explored the myriad other opportunities enabled by the printer PIP, some obvious potential features include "scan to *my* desktop" and "fax to *my* contacts".

² We chose to embed these tags into laminated business cards which were easily carried around in a wallet or pocket

Personalized systems must also **learn and remember things about users** in order to provide personalization. Amazon.com monitors users' shopping behavior in order to learn their tastes, and remembers personal information such as shipping addresses to save users time. In our lab, most people's day-to-day activities revolve around using a single, personal Windows PC in their office. Since we wanted to provide timely access to relevant, personal documents, the PIPs system exploits the document history feature of the Windows operating system, commonly accessed by users through the "Documents" submenu of the "Start" menu. While this menu provides shortcuts to about 10 recently accessed files, the operating system actually maintains a complete archive of file shortcuts from when the user's profile was first created on the machine. Therefore the PIPs system can track what files a user has been accessing without requiring any changes or additional software to be installed on our office PCs. PIPs remembers where to find users' file histories by storing their computer name in a user database. Both embedded and mobile interfaces draw on exactly the same knowledge about users, and require no setup on user machines.

Finally, personalized systems must combine knowledge of users with shared device features to **create a personalized interface for operating the device**. Our three PIPs—presentation PIP, brainstorming PIP, and printer PIP—differ slightly in how they combine knowledge of the user's recent file activity with features of the device they are enhancing. The presentation and brainstorming PIPs both use the file history to provide a "best pick" interface containing a thumbnail of the user's most recently accessed presentation. Choosing the thumbnail displays the presentation—and in the case of our brainstorming PIP, opens it for editing—on the display that the PIP is connected to. In contrast, the printer PIP uses the same underlying file history to provide a slightly different best pick interface containing thumbnails of the six most recently viewed or edited documents of commonly printed document types.³ Choosing one of these thumbnails prints the document on the MFD that the PIP is connected to. In each case, users' recent files are securely accessed through hidden administrative shares, which are typically enabled by default on Windows PCs to allow users to remotely access their documents from other PCs on the network.

In contrast to other solutions which support desktop-based document access anywhere (such as teleporting [12][8]), our approach is device-centered—how can the user's resources be integrated into the device's everyday functions (rather than how can this device be added to the user's desktop). Such integration, when combined with task knowledge, enables us to streamline the user's interaction with a device, and scales to devices with more limited display and interactive capabilities, such as copiers.

³ We analyzed our printer logs for six months to determine which document types to filter through to this interface. These turned out to be Microsoft Office documents, Adobe PDF documents, and web pages.

3.1. Architecture

The PIPs architecture, shown in Figure 3, is web-based. For each shared device, there is a Web server that provides an embedded and portable interface for that device. Interfaces are generated in HTML by server-side ASP scripts that interact with various COM components (e.g., for fetching the user's recent file list) and a "PIP Service" for interacting with local device features (e.g., to run applications *as* the user on the local device).⁴ The system determines which interface variant to return (embedded or portable) based on device characteristics indicated in each HTTP request. The ASP scripts securely impersonate the authenticated user on the network, allowing the PIP to access all the same files accessible to the user when logged into the network via a PC. A shared database is used to associate NT usernames with PIP card IDs and users' document history locations. The PIP Service component provides integration with the actual device, which in our case was always a Windows PC, such as gaining permission to use the currently logged in user's desktop for opening a presentation or printing a document.

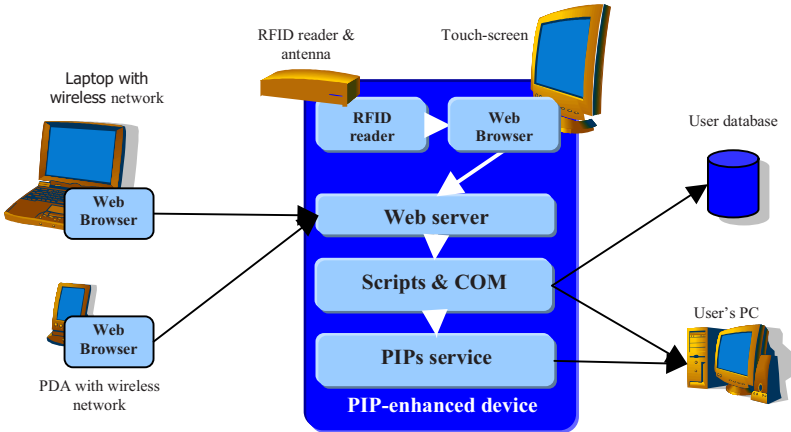


Fig. 3. PIP Architecture: Embedded and portable devices use standard Web browsers to connect to the PIP system via Web servers installed on each PIP-enhanced device. PIPs scripts construct the interface using information from a shared database, resources on the user's PC, and features of the PIP-enhanced device.

3.2. The Embedded Interfaces

The embedded PIP interface is activated when the user approaches the shared device and swipes their smart card over the card reader (Figure 4). In our current implementation we use 32 bytes of rewritable storage on each smart card to store an encrypted version of the user's NT password. The PIP system reads the unique card ID and encrypted password from the card and creates an NT authenticated process which is effectively logged-in as the user. Once authenticated the PIP-enhanced device provides feedback by displaying a splash screen with the user's name and picture indicating they have been recognized.

⁴ This decentralization allowed us to incrementally develop, test and deploy the prototype systems without affecting PIPs in other locations.



Fig. 4. A user logs into an embedded PIP interface by swiping an RFID card.

The PIP Web application then generates the personalized interface by fetching and resolving the shortcuts stored in the user's recent file list on their PC. After a short delay (typically around 6 seconds for 600 recent file shortcuts), the PIP presents a best pick interface with the recent file (or files) the user is most likely to want to use at the PIP-enhanced device (Figure 5 left). The user may then perform a default action (such as “present” or “print”), by pressing on the document's thumbnail using the touch-screen provided by the embedded PIP. Again, files are accessed over the network from their original locations, so users needn't plan ahead or copy files anywhere.

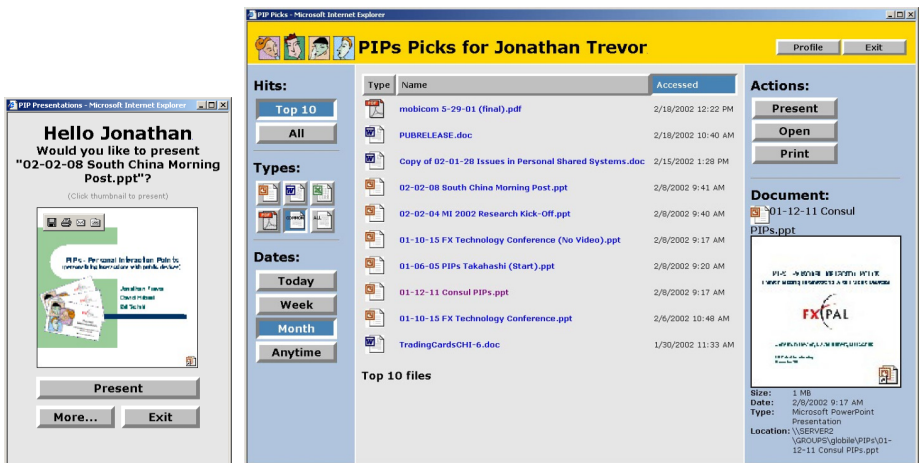


Fig. 5. The embedded PIPs “Best Pick” (left) and “Full” (right) interfaces.

If the best pick interface does not contain the user's desired document, the user can press the “More...” button to bring up the “full” interface (Figure 5 right). This allows the user to access virtually any document (via the device) that they have ever accessed on their office PC. It also allows *other* actions to be performed on files beyond the default action provided by the best pick interface. The left frame allows users to filter and constrain the main file list in the center. Selecting a column in the list sorts the list by type, name, or date. Selecting a document in the list causes the right frame to display relevant actions that may be performed on the file, as well as the name and thumbnail of the file, its size, location, and other details.

3.3. The Portable Interfaces

Our comparative prototypes methodology requires us to provide roughly identical functionality for both design variants, so it was important to design the portable interface to be as similar to the embedded interface as possible. When users access the PIP via their laptop, we provide an interface that is nearly identical to the embedded interface (Figure 5 above), however, users interact using the laptop’s keyboard and mouse instead of the touch screen provided by the embedded interface. When users access the PIP using a smaller portable device, such as a Pocket PC, the smaller screen dimensions required us to make a few cosmetic changes. For example, the file details view (the right-most frame in the full display in Figure 5 above) was separated into its own page.

We decided not to use location sensing technology, and instead presented portable users with a PIP home page for selecting a PIP-enhanced device to connect to. This allowed users to remotely interact with PIP devices without needing to be in the same room with the device.

To use the portable PIP interface, user’s point their device’s Web browser to the PIP home page. Selecting a PIP-enhanced device takes the user to the portable PIP interface for that device. Standard browser authentication is used to prompt for their NT username and password. Once authenticated, the PIP returns a splash screen and proceeds to fetch and resolve the user’s recent file list in the same way it does for the embedded interface. Figure 6a shows the portable version of the best pick interface in Figure 5 (left) on a Compaq iPAQ Pocket PC. Selecting the thumbnail or default action button (e.g. “present”) using the stylus causes that action to be performed on the document at that shared public device. Choosing “More...” provides access to the full interface containing the user’s whole recent file list (Figure 6b), which may then be sorted and filtered as in the embedded interface. Selecting a file from the list causes the actions, thumbnail, and details for a file to be shown (Figure 6c).

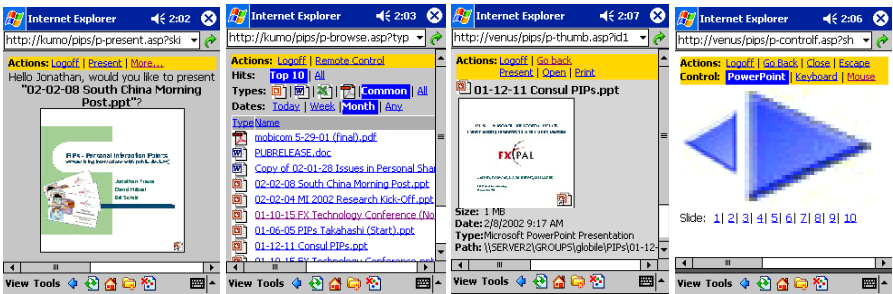


Fig. 6. The “Best Pick” (a), “Full” (b,c), and “remote control” (d) portable interfaces.

The most notable difference between the embedded and portable interfaces comes *after* a file has been selected for presentation, brainstorming, or printing. In the embedded interface the user can use the embedded touch screen, keyboard, and mouse attached to the device to perform subsequent actions on the device. However, in the portable interface, the user may not be close enough to the device to control it

directly, and must therefore remotely control it. Thus, we substitute a “remote control” interface (Figure 6d) to emulate the functions available on the shared device. This functionality resembles that of systems like Pebbles [9]. The impact of this difference is examined further in the following section.

4. Deployment and Observations

The PIPs prototypes were deployed and adopted over the course of several months. The three embedded PIPs were released a year ago, and the portable interfaces released a couple of months later. Ideally we would have deployed the prototype variants simultaneously. However, we believe this didn’t significantly affect our findings since many of our users only started using the system after both variants were available.

The initial months were spent debugging the prototypes and increasing the visibility of the PIP-enhanced devices within the laboratory. Gradually, our user base grew as lab members observed the utility of the system as demonstrated by early adopter usage, primarily in our formal conference room. The trend has been toward increased users *and* usage. Approximately 80% of the research staff is now using PIPs, and so far no one who has used it has subsequently stopped using it. The presentation PIP is used for over half of the presentations given in our formal conference room; the brainstorming PIP is used for nearly all documents accessed in our brainstorming room. Finally, the printer PIP is used for less than one percent of all print jobs. Interestingly, users more frequently used the “Print” action provided by other PIPs (which sends print jobs to the mailroom MFD) than using the printer PIP interface itself. This suggested that users saw value in a personalized interface for the MFD, but one which was integrated with other device interfaces (e.g., connected to the podium PC in our conference room or to the shared display in our brainstorming room), rather than connected to the MFD itself.

During the past year we gathered feedback from early adopters, late adopters, and non-adopters, and recorded incidents in which PIPs failed to operate as expected by our users or us. These eventually became known as “PIPcidents”, and provided much of the fodder for the issues described in the next subsection.

4.1. Issues

Our comparative prototyping methodology illuminated five key issues in the design of personalized ubiquitous systems: usability, utility, availability, trust, and privacy.

4.1.1. Usability

Usability includes learnability, efficiency, memorability, error handling, and user satisfaction [10]. We were primarily concerned with learnability and efficiency, or *general ease of use*, of our alternative interfaces. Our assumption was that usability would decrease with portability due to the limited displays and input mechanisms provided by portable devices.

Our experience confirmed our suspicion. Picking links and scrolling around in the small portable interface required more attention, and was more prone to errors. In contrast, the embedded interfaces, presented on larger touch screens (15" and up), had larger buttons and thumbnails, and allowed users to more easily select files and chose actions.

Another usability difference arose in the separation of the personal interface from the existing shared device interface. In the embedded interface, personalization features were *integrated* with the user interface for controlling the device. As a result, users' existing mental models of how the system worked applied directly to the personal interface without any confusion. However, in the portable interface, the personalization features are provided via an interface that is *separated* from the normal interface for operating the device. This resulted in user confusion. Some people asked: "Do I need to load my presentation onto the iPAQ before going to podium PC?" Apparently the separation of the personal interface from the normal device interface obscured the fact that the podium PC itself (and not the portable device) would be accessing their documents over the network. Also, because the laptop interface is nearly identical to the embedded interface, and since laptop keyboards are similar to the keyboard normally used to control presentations on the podium PC, one of our users *repeatedly* made the mistake of trying to use his *laptop keyboard* to advance through his slides, instead of using the remote control Web interface provided by the PIP system.

In sum, our embedded interfaces were more usable than our portable interfaces. While some of the observed problems might decrease over time as people become more accustomed to using the portable interfaces, the embedded interfaces appeared to have two important advantages: they were easier to use by virtue of being larger, and users' existing mental models "transferred" to the embedded interfaces since they didn't introduce new physical components to the system.

4.1.2. Utility

Utility is functionality that users perceive to be useful. Our comparative methodology called for our embedded and portable interfaces to provide the same basic underlying functionality. Thus, we did not expect to find significant differences in utility between the two approaches.

However the release of the portable interface sparked a change in our practice of giving presentations in the formal conference room. Several presenters liked the ability to remain seated while selecting and controlling their presentation. This provided a much less formal presentation experience for all parties. Following a few presentations of this sort, we attracted new users who—not previously persuaded by the personalization features—saw great utility in remotely controlling presentations.

Thus, our portable interfaces provided added utility over our embedded interfaces. However, variations in utility between the embedded and portable interfaces were also mediated by the situation. In our brainstorming room, the benefits of remote control were lessened since the room is so much smaller than our conference room, and because the usability drawbacks of editing documents via the portable interface outweighed the added utility of remote control.

4.1.3. Availability

We suspected our embedded interfaces would be more *available* than our portable interfaces for numerous reasons. First, users might forget to bring their portable device with them, while embedded interfaces are always available at the shared device. Second, portable power sources and networking are far less reliable than embedded power sources and networking.

Our experiences confirmed our suspicion. For instance, when our CEO decided to use the portable presentation PIP for the first time, we made sure his Compaq iPAQ was fully charged, was connected to the wireless network, and was ready to open the presentation in a single browser click. However, during the meeting, the 802.11b network failed silently, and when he tried to present his document nothing happened. He eventually stood up and swiped his tag at the embedded presentation PIP and gave his presentation. Thus, an intermittent problem in wireless connectivity rendered the portable interface useless. Our embedded interfaces, on the other hand, have proven to be much more reliable.

In sum, our embedded interfaces are significantly more available than our portable interfaces.

4.1.4. Trust

Trust means users believe their personal data is safe from corruption or misuse. Trust is an important issue for many applications, particularly Web sites seeking users' personal information. Research by Ackerman et al. [4] illustrates how users' comfort levels radically change depending on the *types* of information the system has access to. Since our system accesses users' documents (the originals, not copies) containing potentially sensitive information, we were certain trust would be a critical issue.

Surprisingly, very few of our users actually asked any questions about the underlying security of the PIPs system, implying they trusted it implicitly. This is probably because the PIPs system was evaluated within a small and trusted environment: namely our internal, firewall-protected corporate network of approximately 30 users. However, we thought we might see at least subtle differences in user trust depending on whether the interface was embedded or portable. Indeed, numerous non-users, particularly Japanese visitors, raised questions regarding whether users could really trust an interface embedded at a public device, such as an MFD in a 7-Eleven in Japan, to access their sensitive documents. These non-users suggested that people, at least in Japan, might feel more comfortable accessing their personal documents through their cell-phone (a portable interface) than via the MFD itself (an embedded interface).

Unfortunately, we were unable to observe notable differences between user trust of our embedded and portable interfaces. Again, this is most likely due to the fact that our system was evaluated within a trusted environment. However, we believe that trust may indeed vary depending on whether interfaces are embedded or portable, especially for devices in public places.⁵

⁵ It is possible that branding by "trusted" companies may help ease customer concerns (e.g., the Xerox logo on a public copier).

4.1.5. Privacy

Privacy means users are comfortable that their sensitive information will not be revealed to others. Because our system displays interfaces derived from a user's recent file activities, it does reveal information regarding what the user has been doing in his or her private office. We assumed that portable interfaces, being smaller and more personal, would provide users a greater sense of privacy.

The issue of privacy did indeed arise as an issue, particularly in our formal conference room where users are presenting to colleagues from other projects, the whole lab, or visitors. Using the embedded interface provides a thumbnail of the "best guess" presentation both on the podium monitor *and* the main projection screen. Several users started to use the podium's "blank main screen" option to hide the presentation, or their list of files, from others. Some users even suggested that the "blank screen" function might be automatically triggered by the PIP when the user initially logs in, and that the "unblank screen" function might be triggered once the user has selected a document to present. On the other hand, privacy did not come up as an issue for our brainstorming and printer PIPs. The brainstorming PIP provides similar functionality to the presentation PIP, displaying the best pick interface to the entire room. However because the room is used for informal meetings with close associates, users did not perceive a need for privacy, and thus did not notice differences in the embedded versus portable interfaces in this environment.

Thus, privacy varies not only with respect to whether personal interfaces are embedded or portable, but also depending on where the shared device is and what it is used for. People felt uneasy about the privacy of the embedded presentation PIP, while feeling less concerned about the brainstorming and printing PIPs.

4.2. Summary of Findings

Table 1 summarizes differences between the alternative interface approaches across three situations.

Table 1. Summary of issues for personalizing shared ubiquitous devices

			Issues in personalization shared ubiquitous devices				
PIP	Situation	Decision	<i>Usability</i>	<i>Utility</i>	<i>Availability</i>	<i>Trust</i>	<i>Privacy</i>
Presenting PIP	Formal meeting room	Embedded	++	+	++	+	-
		Portable	+	++	+	+	++
Editing PIP	Small meeting room	Embedded	++	+	++	+	+
		Portable	+	+	+	+	++
Printing PIP	Public mail room	Embedded	++	+	++	+	+
		Portable	+	++	+	+	+

Issues of **usability** and **availability** varied more based on design decision (embedded versus portable) and less by situation. Embedded personal interfaces were more *usable* than portable versions irrespective of the situation. Even the portable laptop

interface produced more usability problems than its embedded counterpart due to the separation of the control functions from the underlying device. Embedded interfaces were also more *available* than portable interfaces across situations, being less prone to problems of wireless networks, battery life, and having to carry a portable device with you at all times.

The issues of **utility** and **privacy**, on the other hand, varied not only based on design decision, but also by situation. The portable presentation PIP enabled remote control functionality that was particularly useful in our formal conference room, but less important in the other two situations. Portable interfaces were also better than their embedded counterparts for supporting user privacy. Again, this difference was more marked in our formal conference room, but less marked in our informal brainstorming room and in our mailroom where the likelihood of being overseen was reduced, in part due to the quick nature of activating print jobs. Indeed, it seems that short, less intricate tasks, like printing, were well suited to general portable devices, whereas embedded, custom, interfaces are much better at supporting more complex tasks.

User **trust** did not appear to vary significantly based on design decision or situation in our experiment. However feedback from visitors indicates that people may feel more comfortable accessing personal data via personal portable interfaces, than interfaces embedded in shared public devices.

5. Conclusions

This paper describes an experiment in personalizing shared pervasive devices. The contributions are threefold. First, we present an experimental methodology, *comparative prototypes*, for evaluating ubiquitous systems. Second, we describe the design and implementation of Personal Interaction Points (PIPs), a system that personalizes multi-user document devices in the workplace with recently used files from an individual's desktop computer. The design of PIPs also demonstrates by example how to adopt the comparative prototypes method. Third, we share some of the early lessons we learned from designing, building, and deploying PIPs using this methodology.

Our comparative prototypes methodology gives us a framework for better understanding a ubiquitous computing system. It goes beyond *designing for use* and promotes *designing for evaluation*. The framework produces a matrix of observations and measurements across situations of use that help us to think about ubiquitous computing design.

The experiment and application of the PIPs system in the different situations has reinforced intuitions that designers of ubiquitous systems must carefully consider the *variety* of situations a system will be deployed in and *all* the tasks it is facilitating. We showed that features of near-identical systems deployed in marginally different situations worked in one but not the other.

Our initial experience with the PIPs system indicates that personalization of multi-user document devices can indeed increase ease of use and efficiency. The design decision of how to interact with personalized shared ubiquitous devices, via embedded or portable interfaces, is an important one and involves trade-offs. Our matrix indicates differences with respect to five issues between embedded and portable interfaces across three situations.

Others may use this matrix as a point of reference in making their own design trade-offs, or may apply the methodology to produce their own matrix featuring different design decisions, situations, and issues.

References

1. Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, Mike Pinkerton. Cyberguide: a mobile context-aware tour guide, *ACM Wireless Networks*, 3, pp. 421-433, 1997.
2. Gregory D. Abowd. Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. *IBM Systems Journal, Special issue on Pervasive Computing*, Vol. 38, No. 4, pp. 508-530, October 1999.
3. Gregory D. Abowd, Elizabeth D. Mynatt and Tom Rodden. The Human Experience. *IEEE Pervasive Computing*, March, pp.48-57, 2002
4. Mark S. Ackerman, Lorrie F. Cranor, and Joseph Reagle. Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences. In Proceedings of ACM Conference on Electronic Commerce, Denver, CO. pp.1-8, 1999.
5. Jonathan Grudin. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organisational Interfaces. Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88) , 1988
6. Mik Lamming, Marge Eldridge, Mike Flynn, Chris Jones and David Pendlebury. Satchel: providing access to any document, any time, anywhere. *ACM Transactions on Computer-Human Interaction*, Vol. 7, No. 3, 2000.
7. Jennifer Mankoff and Bill Schilit. Supporting Knowledge Workers Beyond the Desktop with Palplates. In Proceedings of CHI 97, ACM Press, March 22-27 1997, pp. 550-551.
8. Microsoft. Microsoft Windows 2000-based Roaming User Profiles Microsoft Windows 2000-based Roaming User Profiles, Microsoft Corp., Details available at <http://www.microsoft.com/technet/showcase/deploy/roamprof.asp>
9. Brad A. Myers, Herb Stiel, and Robert Gargiulo. Collaboration Using Multiple PDAs Connected to a PC. Proceedings CSCW'98: *ACM Conference on Computer-Supported Cooperative Work*, Seattle, WA. pp. 285-294. November 14-18,1998.
10. Jacob Nielsen. Usability Engineering. Academic Press/AP Professional, Cambridge, MA, 1993
11. Shankar R. Ponnekanti, Brian Lee, Armando Fox, Pat Hanrahan, and Terry Winograd ICrafter: A Service Framework for Ubiquitous Computing Environments Proceedings of Ubicomp 2001, September 30-October 2, 2001
12. Tristan Richardson, Frazer Bennett, Glenford Mapp, and Andy Hopper. Teleporting in an X Window System Environment, *IEEE Personal Communications Magazine*, Vol. 1, No. 3, pp 6-12, 1994.
13. Jean Scholtz and Asim Smailagic. Workshop on Evaluation Methodologies for Ubiquitous Computing, Ubicomp 2001. Report published in *SIGCHI Bulletin*, January/February 2002.
14. Asim Smailagic, Daniel P. Siewiorek, Joshua Anhalt, Francine Gemperle, Daniel Salber, Sam Weber, Jim Beck, Jim Jennings. Towards Context Aware Computing: Experiences and Lessons Learned, *IEEE Journal on Intelligent Systems*, Vol. 16, No. 3, June 2001
15. Richard M. Stallman. EMACS the extensible, customizable self-documenting display editor. *ACM SIGPLAN SIGOA symposium on Text manipulation*, Portland, OR. pp. 147 □ 156, 1981
16. Texas Instruments, Radio Frequency Identification Systems, <http://www.ti.com/tiris>
17. Richard van de Sluis, Berry Eggen, John Jansen, and Han Kohar. User Interface for an In-Home Environment. Proceedings of the Eighth IFIP TC.13 Conference On Human-Computer Interaction, 2001.
18. Michael Twidale, David Randall and Richard Bentley. Situated evaluation for cooperative systems. Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 94), 1994.
19. Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The Active Badge Location System, *ACM Transactions on Information Systems*, Vol. 1, No. 10, pp 91-102, 1992.