Personalizing shared ubiquitous devices

David M. Hilbert and Jonathan Trevor
Fuji-Xerox Palo Alto Laboratory, Inc.

Everywhere we go, we are surrounded by shared devices: TVs, stereos, and appliances in the home; copiers, fax machines, and projectors in the office; phones and vending machines in public. Because these devices don't know who we are, they provide the same user interface 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, personalization pervades the Web, and is beginning to play a role in everyday devices such as cars and DVRs.

This article describes two years of experience with a research prototype for personalizing shared workplace devices such as projectors, public displays, and multi-function copiers. The system combines users' networked resources—or "personal information clouds"—with device-specific user interfaces for performing common device tasks. We developed and compared personal interfaces that are embedded (i.e., integrated or co-located with the shared device) and portable (i.e., accessible via personal devices such as mobile phones and PDAs).

Our experience indicates that a little personalization can go a long way toward improving user friendliness, efficiency, and capabilities of shared document devices, helping them "weave themselves into the fabric of everyday life" [1]. We also gained important insights into subtle differences between embedded and portable approaches to ubiquitous computing systems.

Figure 1. The BMW 7 Series remembers your seat, mirror, and steering wheel settings and recalls them automatically when you use your unique key to enter the car (Left). TiVo DVRs learn your TV viewing preferences and automatically record TV shows you like (Right).

Personalization and shared devices

Personalization has a rich history in computer science and has recently enjoyed a revival [2]. Personalization permeates the literature in human-computer interaction, computer-supported collaborative work, operating systems, and the World Wide Web. Emacs, for example, is well known for being end-user customized for a wide range of tasks, from program editing to file browsing and news reading [3]. Olivetti Research Laboratory pioneered an early system for "teleporting" one's personal computing environment from one device to another as a way of personalizing shared terminals [4]. More recently, Microsoft has embraced so-called "smart" menus in Office and Windows that adapt to their users.

Personalization also plays an important role in e-commerce Web sites and is beginning to find its way into consumer products. Amazon.com uses personalization to recommend products and simplify ordering, so users can just *shop.* Likewise, the BMW 7 Series remembers seat, mirror, and steering wheel settings so drivers just enter the car and *drive.* Finally, TiVo DVRs automatically record shows so viewers can just *watch.* Our work is situated in this tradition, but focuses on workplace document devices, such as photocopiers.

Designing for people

Like others in the HCI and CSCW research communities, we believe designing for people begins with observation. So we began our exploration of shared device personalization by observing our own colleagues interacting with three shared document devices in our workplace: the presentation PC in our formal conference room, a large plasma display in our brainstorming room, and the multi-function copier in our mailroom.

Figure 2. Three shared devices in our laboratory: the presentation PC in our formal conference room (left), a large plasma display in our brainstorming room (center), and the multi-function copier in our mailroom (right).

In each case we noticed users engaging in peripheral activities that took their attention away from *presenting*, *brainstorming*, and *printing*. In each case we identified ways in which personalization could help reduce these secondary activities so people could focus on their primary goals.

The podium PC

The speakers' podium in our formal conference room (Figure 2 left) is equipped with a networked PC that drives two displays: a small monitor in the podium itself and the room's main projection screen. Presenters take turns using the podium PC to show their presentations on the main screen.

We noticed users spent considerable time, sometimes several minutes, searching for their presentations on the network. This resulted in awkward moments while presenters located their files, breaking up meeting flow, particularly in meetings with multiple presenters. To avoid this, some users began copying their presentations to shared folders on the podium PC beforehand. However, this required pre-planning and discipline that not all our colleagues shared, and led to versioning problems when people edited *copies* of their presentation on the podium PC or in their office.

In short, the podium PC, which is essentially an interface to our projector, was distracting users from their primary goal. This led us to the idea of a "personalized projector" that would seamlessly locate and open users' presentations so they could focus their attention on *presenting.*

The plasma display

Our brainstorming room (Figure 2 center) is a smaller room that accommodates about eight to 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 brainstorms.

However, users rarely used the plasma display. 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 users to copy their documents to a shared laptop (if one was available) and to connect it to the display prior to brainstorming.

In short, the plasma display required users to engage in too many non-brainstorming activities to serve as a useful brainstorming tool. Thus, we envisioned a "personalized brainstorming display" that would streamline the activity of opening personal working documents on the plasma display so that users could focus their attention on *brainstorming.*

The multi-function copier

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 is used by nearly everyone daily.

People in our lab most frequently used the MFD to print documents from their offices. While the MFD serves this purpose well, we noticed some situations in which it could be improved. For instance, users wanting to print documents during meetings sometimes had to go back to their office to initiate jobs even though the meeting room was closer to the MFD than to their office. On several occasions, users returned to their office to re-initiate print jobs when output did not meet their expectations. Finally, some users wanting to print "sensitive" documents (not knowing how to use the MFD's print-and-hold function) started jobs from their office, and then ran to the mailroom to pick them up before others could see them.

This led us to consider what it would be like to have a "personalized MFD" that would allow users to print their personal documents while standing in front of the MFD. Users wouldn't always need to be in their offices to start print jobs, and they could also access other personal features such as: "scan to my desktop" instead of scanning to a shared network folder; "fax to my contacts" instead of looking-up the fax number and copying and entering it into the MFD; and the ability to re-use their personal copier preferences and fax history.

Personal information clouds

As we reflected on how to improve the user experience of these shared devices, we immediately thought of "personal information clouds," a concept that others have found useful in the past [5][6]. Personal information clouds could follow us wherever we went, and ease the flow of information between the many document devices we interact with on a daily basis. Our clouds would contain data we touched throughout the day. For instance, editing a Word document on my desktop, or a PowerPoint presentation on my laptop, would populate my cloud with these documents. Furthermore, our clouds would become populated through the use of *any* computing device, including multi-function copiers. So if I copied or scanned a document, it would appear in my cloud. If I faxed a document to one of my contacts (also stored in my cloud) my personal fax history would be updated in my cloud. We could use these information clouds to personalize multi-user document devices. For instance, when I walk up to the podium PC, I could access my cloud to quickly launch a recently edited presentation on the projector.

Figure 3. Personal information clouds are populated by data we touch throughout the day (left). They are also populated when we use non-traditional computing devices, such as MFDs (center). Our information clouds can then be used to personalize and augment the capabilities of existing shared devices (right).

With this concept in mind, we set out to design a system for personalizing multi-user document devices. We saw this system as an exploration into some of the benefits personalization could bring to the anonymous interfaces and functionality found in copiers, fax machines, and printers (such as Fuji Xerox's MFDs) that are so common in hotel business centers, convenience stores, and copy shops. We reasoned that if these devices were "aware" of their users, they could automatically access users' personal networked resources and integrate them into the user interface for controlling the device.

So, for instance, we could use a public MFD to print our recently accessed PC documents or to route faxes using our personal address books.

We immediately thought of numerous personalization features, but wanted to focus our effort on a small set of functionality with a large payoff for users. This was particularly important since we wanted to secure real use in order to observe the effects of personalization on real users. Thus, we decided that integrating personal computer file access with a shared device's functions would be a good starting point. Our system embodies this by giving "smart" access to people's file 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 to the function types of devices. So for a conference room projector, the system automatically selects PowerPoint presentations from the user's information cloud as the documents the user is most likely to want to present.

The main distinguishing feature of our approach is combining resources from users' personal information clouds with device-specific interfaces for performing common device tasks. So instead of using teleporting [4] or virtual network computing (VNC) [7] to remotely access your desktop to find and start a presentation at the podium PC, you use a special interface for showing presentations that links directly to your recently edited presentations.[1]

Designing for evaluation

Once researchers have observed users and envisioned how technology might support their activities, we often build and deploy working prototypes to observe them in use. This has produced numerous insights regarding systems and their effects on users and the social contexts in which they are embedded. However in deploying real world systems, we are forced to choose between myriad design alternatives, some of which may 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?

In our case, we immediately faced a fundamental question in ubiquitous computing: whether to use embedded or portable interfaces? An embedded approach would integrate personalization with an already existing (or co-located) user interface for controlling the device. A portable approach would provide a personal interface via portable devices, such as mobile phones or PDAs.

[1] Our approach is reminiscent of Intel's personal servers [8][9]. Personal servers are small but powerful wearable hardware (data storage and processing) that users can access via displays and other devices in the environment. While personal servers offer an attractive way of "physically" instantiating users' personal information clouds, we created virtual clouds by connecting people to their resources stored on various networked PCs and file servers.

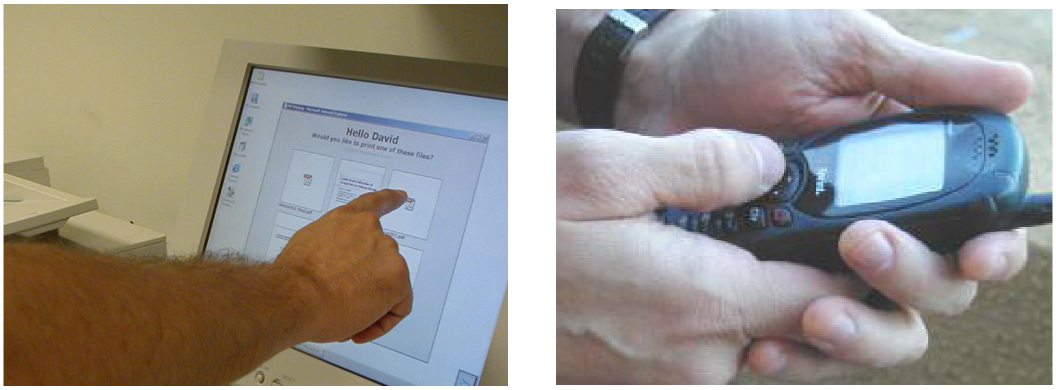

Figure 4: Two key design alternatives for ubiquitous systems: embedded (left) or portable (right) interfaces.

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. Would users feel uncomfortable 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?

However, evaluating ubiquitous systems is hard, and has captured the attention of others in the research community [10]. These investigators, like others in CSCW [11][12], 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). As Abowd, Mynatt, and Rodden put it: "Deeper empirical evaluation results cannot be obtained through controlled studies in [the] traditional, contained usability laboratory. Rather, the requirement is for real use of a system, deployed in an authentic setting" [13]. So we needed to devise our own evaluation technique that would be similar to a comparative laboratory experiment, but that would allow us to observe effects of our design decisions in relatively unconstrained, real world use. Our approach can be summarized as follows:

1. Observe users
2. Design a system grounded in observation
3. Build a system that embodies more than one design alternative
4. Deploy the system in various situations
5. Compare the design alternatives across the deployment situations

Steps 1 and 2. We had already observed users of shared document devices in our organization. We used our observations to design a system that would address real problems and fit in with existing practices so that people would actually use it. We also strove to make the underlying functionality as simple as possible, but at the same time compelling enough to produce real usage.

Step 3. The next step was to build our design alternatives. So we constructed a system that users could access via an embedded interface co-located with the shared device, or via a portable interface hosted on a portable device. We made sure both variants provided the same basic underlying functionality so that our comparison could focus on our main design alternatives.

Step 4. Since ubiquitous systems are often deployed in a variety of situations, we need to evaluate them in various settings to help answer the question: are observed effects linked to a particular situation or are they more general? We had already identified three shared document devices (the podium PC, plasma display, and MFD) located in three different locations (the conference room, brainstorming room, and mailroom). So we deployed our system in each of these situations.

Step 5. Finally, we observed system use to compare and contrast the design alternatives across the various situations. We collected a combination of qualitative and quantitative data including: observation notes, unsolicited E-mail and comments from users, informal interviews of adopters and non-adopters, and usage logs. We used the data to populate a matrix of observations—with design alternatives along one axis, and deployment situations along the other—to help determine whether observed effects varied depending on the design alternatives, the varied situations, or both.

Our evaluation technique goes beyond traditional *designing for use* (most notably, steps 3 and 4), and promotes *designing for evaluation.* The resulting observation matrix provides a framework for better understanding ubiquitous systems.

The personal interaction points system

Our process of designing for evaluation led us to create the personal interaction points (PIPs) system. The PIPs system is a web-based application that generates embedded and portable interfaces for personalizing the projector in our conference room, plasma display in our brainstorming room, and MFD in our mailroom.

In our conference room, we updated the podium PC with a touch-screen display. Our "presentation PIP" provides users with a personal interface for showing their most recently accessed presentations on the main projection screen. No file copying or synchronization is required as documents are securely accessed over the network from their original locations.

In the brainstorming room—rather than relying on users to connect laptops to the display—we provided a networked PC and added a touch screen overlay and wireless keyboard. Our "brainstorming PIP" provides users with a personal interface for viewing and editing their working documents on the plasma display. Users may also create new files for note taking, and all document updates are automatically saved back to their original locations.

Finally, in our mailroom, we added a 15-inch touch-screen display connected to a networked PC running the PIP software, which we hid in a cabinet next the MFD. Our "printer PIP" provides users with a personal interface for printing their recent documents on the MFD.

The embedded interface

Each embedded PIP consists of a touch screen on or near the shared device and a Radio Frequency Identification (RFID) card reader. The embedded interface is activated when the user approaches the shared device and swipes their ID card over the card reader. The system reads the users' encrypted password from the card and starts an NT authenticated process that runs as the user. Users who forget their cards can also login by entering their username and password using an optional keyboard.

Figure 5. A user logs into an embedded PIP interface by swiping their RFID card.

The PIP web application then generates the personal interface by fetching and resolving the shortcuts stored in the user's recent file list on their PC. 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 6 left). The user may then perform a default action (present for the projector, open for the plasma display, and print for the printer), by selecting the document's thumbnail using the touch-screen provided by the embedded PIP. Files are accessed over the network from their original locations, so users needn't plan ahead or copy files anywhere.

Figure 6. The embedded "best pick" interface suggests the document (or documents) the user is most likely to want to use on the PIP-enhanced device (left). Selecting the thumbnail causes a device-specific action to be performed on the document (present for the projector, open for the plasma display, and print for the printer). The user may also select "More" to explore other resources contained in their information cloud (right). Here users can flexibly filter and sort their clouds. The

left frame allows users to filter the file list in the middle frame. Selecting columns in the middle frame sorts the list, and selecting a file causes the right frame to display actions applicable to the selected file, as well as its thumbnail, size, and other attributes.

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 6 right). This allows the user to access virtually any document (via the device) that they have ever accessed on their office PC.

The portable interface

For the portable interface, users point their portable Web browsers at the PIPs home page. Selecting a PIP-enhanced device from the home page activates the portable PIP for that device. Standard browser authentication is used to login users. Once authenticated, the PIP application fetches and resolves the user's recent file list in the same way it does for the embedded interface.

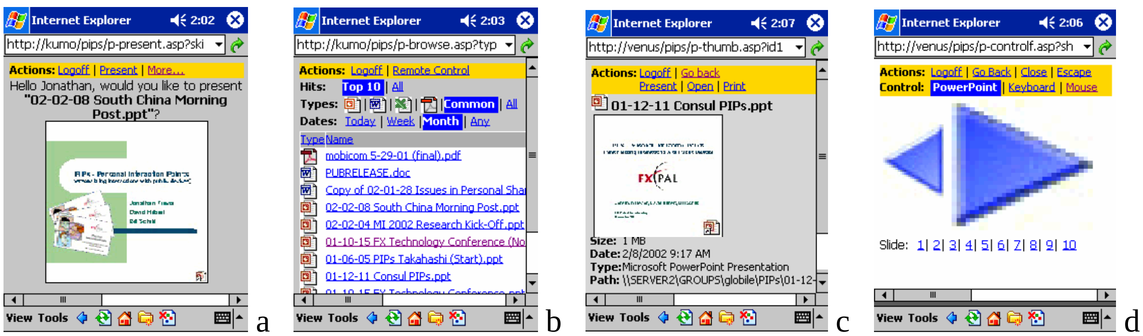

Figure 7. The portable "best pick" interface suggests the document (or documents) the user is most likely to want to use on the PIP-enhanced device (a). Selecting the thumbnail or "Present" with the stylus causes the document to open on the PIP-enhanced device. The user may also select "More" to explore other resources contained in their information cloud (b). Selecting a file displays the file details page (c). Finally, a remote control interface allows users to control the PIP-enhanced device from afar (d).

While we strove to keep the embedded and portable interfaces as similar as possible, we were forced to miniaturize the interface and make other minor modifications for portable devices with small displays, such as Pocket PCs. For instance, the file details view (the right-most frame in Figure 6 right) was separated into its own page (Figure 7c). However, the most notable difference comes after a file has been selected for presentation, brainstorming, or printing. Users of the embedded interface can use the touch screen, keyboard, and mouse attached to the document device to perform subsequent actions on the device. However, in the portable case, the user may not be close enough to the shared device to control it directly, so we created a "remote control" interface (Figure 7d) to emulate the functions available on the shared device.

Evaluation

We deployed the personal interaction points system over the course of several months. We released the embedded interfaces three years ago and portable interfaces a couple months later. Ideally we would have deployed the system 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.

We spent the initial months debugging the prototypes and increasing the visibility of the PIP-enhanced devices within the laboratory. Gradually, we gained new users as lab members observed the utility of the system as demonstrated by early adopters, primarily in our formal conference room.

The trend has been toward increased users and usage. About three quarters of our approximately thirty-person research staff is now using PIPs, and very few people who have used it have subsequently stopped using it. The presentation PIP is used for over half the presentations given in our formal conference room. The brainstorming PIP is used for nearly all documents accessed in our brainstorming room. The printer PIP is rarely used and has since been decommissioned.[2]

Lessons learned

Our results reinforce the fact that designers must carefully consider subtle differences in the situations in which their systems are deployed. Our three prototypes delivered nearly identical functionality but in three different situations. Two of the situations—the formal conference room and the informal brainstorming room—called for a nearly identical interface for quickly and easily opening users' recently edited desktop presentations. However, small differences in the rooms, people, and tasks led to significant perceived differences in usability, utility, and privacy. In the following subsections we summarize lessons learned regarding embedded versus portable interfaces for personalizing shared ubiquitous devices.

Advantages of embedded interfaces

In our experience, embedded interfaces are more usable, available, and simpler to implement than portable interfaces.

Embedded interfaces are more **usable** than portable interfaces, due primarily to their larger displays (15" and up in our case) and flexible input mechanisms (touch screen and optional keyboard and mouse). Usability issues were most noticeable for complex tasks, such as document editing in our brainstorming room. The portable interfaces also confused users and altered the user experience, particularly when users were accustomed to interacting directly with the shared device. One of our users asked: "Do I need to load my presentation onto the Pocket PC before presenting in the conference room?" Apparently the separation of the personal interface from the underlying shared device obscured the fact that the system always opens users' documents over the network, regardless of whether they use the embedded or portable interface. Another user reported that with the embedded interface, "you feel you have a real relationship with the device, but with the portable [interface], you feel you have a relationship with the portable device rather than the actual device." He concluded: "I think of them as two completely separate applications."

Embedded interfaces were also more **available** than their portable counterparts. We observed the portable interfaces suffering from nearly every imaginable availability issue

[2] People in our organization didn't use the printer PIP often since they were typically in their office, or not far from it, when they needed to print. A personalized MFD would clearly be more useful in public locations or in large organizations where users aren't always within a few steps of their office PCs.

ranging from batteries dying and wireless network failures, to users forgetting their devices in their offices.

Finally, embedded interfaces were slightly **simpler to implement** than the portable. In both cases, we faced challenges integrating personalization with existing device hardware. We worked around this by developing PC "proxy" interfaces to drive each device (projector, plasma display, and printer). In the embedded case, once users select a file they can continue to control the device using the device's existing interface. But since portable users may not be close enough, portable interfaces must provide additional remote controls. These remote interfaces are typically more difficult to use than the hardware they are emulating and may require significant additional effort to develop.

These observations taken together suggest designers should consider incorporating personalization into already existing embedded interfaces—to the degree possible—rather than creating new portable interfaces, especially when shared devices support complex tasks.

Advantages of portable interfaces

On the other hand, portable interfaces have an edge in terms of remote control and privacy.

Users found the **remote control** capabilities of the portable interfaces to be quite useful, particularly for simple, on-going tasks in large spaces, such as advancing through slides in our conference room. However, they found remote control to be far less compelling for complex tasks (such as document editing) and one-shot interactions (such as printing) in smaller spaces.

Similarly, users appreciated the **privacy** of the portable interfaces, particularly in the conference room setting. This is because they could access their information clouds via a small private display before presenting, instead of using a larger display that others in the room could see. However, this only came up in the conference room where the mood is more formal and speakers are presenting to colleagues from other projects, the whole lab, or visitors. It was not an issue in informal settings, such as our brainstorming room, or for quick interactions in low-traffic areas, such as printing in our mailroom.

The observed advantages of embedded and portable interfaces taken together suggest designers should consider hybrid solutions that allow users to interact with portable interfaces for remote control and highly sensitive tasks, and embedded interfaces for more complex tasks. For instance, it would make sense for users to use a portable interface (e.g., their mobile phone) to select and transmit personal resources to a public device (such as a convenience store MFD), at which point they could switch to the public device's embedded interface to complete their tasks (e.g., to adjust printing and output options).

Other observations

We initially expected **trust** would be a significant issue, particularly since our system accesses and transfers users' personal documents containing potentially sensitive information. We also suspected some users would perceive personal interfaces to be inherently more trustworthy than public interfaces. However, surprisingly few users asked any questions regarding the underlying security of the system, implying they trusted it implicitly. This is probably because our system was evaluated within a small and trusted environment: namely our internal, firewall-protected corporate network of approximately 30 users. As a result, user trust did not vary depending on design alternative or situation in our experiment. However several Japanese visitors (who were not users) hypothesized that subscribers to a mobile printing service would feel more comfortable selecting their personal documents through their cell-phone (portable interface) than via a public MFD touch screen (embedded interface).

We also gained a deeper appreciation for the delicate balance between **proactivity** and user control [9]. In the conference room and brainstorming room, the "best pick" interface proactively recommends the user's most recently accessed desktop presentation as the file the user is most likely to want to present. This approach works remarkably well and saves time in practice, guessing correctly over 75% of the time in the conference room and over 50% of the time in the brainstorming room. Nonetheless, some users requested more user control to manually designate a particular presentation to be their "best pick", so we deployed a feature that allows users to pick a presentation to be their "active" presentation for the day, several days, or indefinitely. The feature was used several times in the conference room, but never in the brainstorming room. This makes sense since users want to avoid browsing their information clouds publicly in formal settings. But it taught us that recommendation accuracy is an insufficient metric for assessing the value of proactivity: users wanted more control in exactly the situation in which the system's proactivity was most accurate.[3] With the exception of extremely formal presentations to top executives, most users continue to rely on the system to make a best guess, and simply use the "More" button when it doesn't guess correctly.

Conclusions

A little personalization can go a long way toward improving the user friendliness, efficiency, and capabilities of shared ubiquitous devices. We transformed the user experience of three shared document devices in our lab. Now when we use the podium PC, in our conference room, we no longer see it as a general PC for finding and opening presentations. We view it as a specialized device that allows us to swipe our ID card to begin presenting. The former tasks of locating and opening documents have receded into the background so we can focus on *presenting.*

Many researchers in ubiquitous computing implement either wholly embedded or wholly portable interfaces. Since we were unsure about which approach to take, we embodied both alternatives in our system and deployed it in a variety of situations to compare the two. Our technique led to a system that is still in use three years after its introduction.

Today our embedded interfaces are far more popular than their portable counterparts. This could change as wireless devices and networks become more dependable and pervasive, and as users become more accustomed to using them. However, our experience suggests that embedded (or hybrid) interfaces may be inherently more suitable

[3] Ironically, in giving up proactivity in *recommending* documents, we gained proactivity in that we could automatically *open* users' "active" documents for them. As a result, users don't need to wait for the system to resolve their recent files shortcuts to make a best guess, streamlining the process by several seconds.

for particular shared devices, namely, those that support complex tasks. If we had simply decided to go with a portable approach, our system would not have achieved the usage it did, and therefore we would not have gained the knowledge we have. We believe our experimental technique is applicable to other HCI, CSCW, and ubiquitous systems research.

At the end of the day, we like most researchers in our field feel rewarded when we produce systems that actually improve the user experience. The PARCTAB researchers felt they had achieved this goal when their users began complaining they couldn't take their tabs home with them, beyond the reach of PARC's infrared network [14]. Similarly, our users became accustomed to their personalized document device interfaces, and have begun wishing they could take their personal information clouds with them beyond our corporate network. As a result, we have begun modifying our architecture to extend its reach. Soon we will be able to access our personal resources through personal document devices spread across our geographically distributed organization. Perhaps one day this will extend to devices in public places as well.

Acknowledgments

We would like to thank our colleagues at FXPAL for their patience in using our system as it evolved, and for their invaluable feedback. We'd also like to thank Bill Schilit for his inspiration and guidance.

References

1. Mark Weiser. The computer of the 21st century. Scientific American, 265(3):66-75, September 1991.
2. Doug Riecken, Special Issue on Personalization, Communications of the ACM, 43(8). Aug. 2000.
3. Richard M. Stallman. EMACS the extensible, customizable self-documenting display editor. Proceedings of ACM SIGPLAN SIGOA Symposium on Text manipulation, 1981. pp. 147-156.
4. Tristan Richardson, Frazer Bennett, Glenford Mapp, and Andy Hopper. Teleporting in an X Window System Environment, IEEE Personal Communications Magazine, 1(3), 1994.
5. Mik Lamming and Mike Flynn. Forget-me-not: Intimate Computing in Support of Human Memory. In Proceedings of the '94 Symposium on Next Generation Human Interface. Feb, 1994
6. Bradley Rhodes and Thad Starner. The Remembrance Agent: a continuously running automated information retrieval system. In Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96). London, UK, 1996; 487-495.
7. Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper. Virtual Network Computing. IEEE Internet Computing, 2(1), Jan-Feb 1998, pp. 33-38.
8. Roy Want, Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar, John Light. The Personal Server: Changing the Way We Think about Ubiquitous Computing. Proceedings of the Fourth International Conference on Ubiquitous Computing (Ubicomp 2002), pp. 194-209.

9. Roy Want, Gaetano Borriello, Trevor Pering, Keith I. Farkas. Disappearing Hardware. IEEE Pervasive Computing, 1(1), Jan-Mar 2002, pp. 36-47.
10. Jean Scholtz and Asim Smailagic. Workshop on Evaluation Methodologies for Ubiquitous Computing. Proceedings of the Third International Conference on Ubiquitous Computing (Ubicomp 2001). Report published in SIGCHI Bulletin, Jan-Feb 2002.
11. Jonathan Grudin. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88), 1988.
12. Michael Twidale, David Randall and Richard Bentley. Situated evaluation for cooperative systems. Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '94), 1994.
13. Gregory D. Abowd, Elizabeth D. Mynatt and Tom Rodden. The Human Experience. IEEE Pervasive Computing, 1(1), Jan-Mar 2002, pp. 36-47.
14. Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis and Mark Weiser. The ParcTab Ubiquitous Computing Experiment. In Mobile Computing, H. F. Korth and T. Imielinski, eds., Kluwer Academic Press, 1996.

Authors

David M. Hilbert is a senior research scientist at FX Palo Alto Laboratory. His research interests lie in the design and evaluation of practical HCI, CSCW, and ubiquitous computing applications. He received a PhD in information and computer science from the University of California, Irvine. He is a member of IEEE, ACM, and Phi Beta Kappa Society. Contact him at hilbert@fxpal.com.

Jonathan Trevor is a senior research scientist at FX Palo Alto Laboratory, where he works in ubiquitous systems and computer-supported cooperative work. His current research interests are in the development of readily accessible groupware and human-computer interaction applications across a range of technologies and platforms. He received a PhD in computer science from the University of Lancaster. Contact him at trevor@fxpal.com.