

# Embedded versus portable interfaces for personalizing shared ubiquitous devices

David M. Hilbert and Jonathan Trevor

FX Palo Alto Laboratory  
3400 Hillview Avenue  
Palo Alto, CA 94304 USA  
{hilbert, trevor}@fxpal.com

## Abstract

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 paper describes a system for personalizing workplace document devices—projectors, public displays, and multi-function copiers—that has been in use for over two years in our organization. We compare user interfaces that are *embedded* (i.e., integrated or co-located with the shared device) versus *portable* (i.e., accessible via portable devices such as mobile phones or PDAs). We summarize lessons learned for others designing interfaces for shared ubiquitous devices.

## 1 Personalization and everyday devices

In the Web's 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 uses personalization to expedite ordering and to suggest products you may want to buy, based on past purchases and purchases of others with similar interests. Instead of attending to the details of entering billing and shipping information and locating products, these tasks recede into the background so you can focus on *shopping*.

Indeed, this trend is beginning to affect *everyday* shared devices, such as cars and televisions. The BMW 7 Series car remembers drivers' seat, mirror, and steering wheel settings and recalls them automatically when family members use their unique keys to enter the car. Rather than fussing with seats, mirrors, and steering wheel, you just *drive*. TiVo digital video recorders learn your TV viewing habits and automatically record shows for you. Instead of searching through listings and manually setting-up recordings, you just turn on your TV and *watch*. Our work is situated in this tradition, but with a focus on workplace document devices, such as photocopiers.

## 2 Embedded versus portable interfaces

In designing a system to personalize workplace document devices, we immediately faced a fundamental question in ubiquitous computing: whether to use embedded or portable 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 personal resources through an interface embedded in 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? While we had numerous

intuitions, we wanted to compare embedded versus portable interfaces to observe their relative strengths and weaknesses in practice.

### **3 Personalizing workplace document devices**

We began our exploration of shared device personalization by observing our own colleagues interacting with three document devices in our workplace: the projector in our formal conference room, a large plasma display in our brainstorming room, and the multi-function copier in our mailroom. 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.

For the projector in our conference room, we envisioned a personal interface for seamlessly locating and opening users' recently edited presentations. This would allow users to focus on presenting instead of spending time locating their documents on the network using the podium PC. For the plasma display in our brainstorming room we envisioned a personal interface for streamlining the activity of locating and opening users' working documents. Thus, users could focus on brainstorming instead of copying documents from their office PCs to laptops and then connecting laptops to the plasma display. Finally, for the multi-function device (MFD) in our mailroom we envisioned a personal interface so users could print personal documents while standing in front of the MFD. This way people wouldn't always need to be in their offices to start print jobs, and they could access other personal features such as: "scan to my desktop" instead of scanning to a public network folder; "fax to my contacts" instead of looking-up the fax number and copying and entering it into the MFD; and re-using their personal copier preferences and fax history.

#### **3.1 The Personal Interaction Points system**

The Personal Interaction Points (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. 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. PIPs 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 is that it matches file types to the function types of devices. So for a conference room projector, the system automatically selects PowerPoint presentations from the user's file history as the documents the user is most likely to want to present.

The main distinguishing feature of PIPs is combining users' networked resources—or "personal information clouds"—with device-specific interfaces for performing common device tasks. So instead of remotely accessing your desktop (Richardson et al. 1998) to locate and start a presentation on the presentation PC, you use a special interface for showing presentations that links directly to your recently edited presentations. We now briefly describe the embedded and portable interfaces. For more details see (Trevor et al. 2002).

##### *3.1.1 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'



**Figure 1.** The “best pick” interface suggests the document (or documents in the case of the printer) 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 in their file history (right). 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, its name and thumbnail, and other file details.

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.

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 1 left). The user may then perform a default action (such as present, open, or print), 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. 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 1 right). This allows the user to access virtually any document (via the device) that they have ever accessed on their office PC.

### 3.1.2 The portable interface

For the portable interface, users point their portable Web browsers at the PIPs homepage. Selecting a PIP-enhanced device from the homepage 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.

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 “full” interface (Figure 1 right) was split into two pages: one for the main file list and another to show selected file details. 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 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 simple “remote control” interface to emulate the functions available on the shared device.

## 4 Lessons Learned

Over two years have passed since we deployed the PIPs system. Today, about three quarters of our approximately thirty-person research staff actively uses it, and so far no one who has used it has 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.<sup>1</sup> In the following subsections we summarize lessons learned regarding embedded versus portable interfaces for personalizing shared ubiquitous devices.

### 4.1 Embedded advantages

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 users' documents are always opened 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 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 MFD). 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.

### 4.2 Portable advantages

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

---

<sup>1</sup> 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.

Users found the **remote control** capabilities of the portable interfaces to be quite useful, particularly for simple and 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 single-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. But again, this was only noticed in the conference room where the mood is more formal and speakers are presenting to colleagues from other projects, the whole lab, or visitors. Privacy 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, users could use a portable interface (e.g., a mobile phone) to select and transmit personal resources to a public device (e.g., an 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).

## 5 Conclusions

A little personalization can go a long way toward improving the user friendliness, efficiency, and capabilities of shared document devices. We transformed the user experience of three shared document devices in our lab. Now when we use the presentation PC, 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*.

Ubiquitous computing researchers typically 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 more than two 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 for particular shared devices, namely, those that support complex tasks requiring complex interfaces. 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, and encourage others to decide for themselves (Trevor et al. 2002).

## 6 Acknowledgements

We would like to thank Bill Schilit for his inspiration and guidance.

## References

- Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, Andy Hopper (1998). Virtual Network Computing. *IEEE Internet Computing*, 2(1), 33-38.
- Jonathan Trevor, David M. Hilbert, and Bill N. Schilit (2002). Issues in personalizing shared ubiquitous devices. *Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp 2002)*.