



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Interacting  
with  
Computers

Interacting with Computers 17 (2005) 251–264

[www.elsevier.com/locate/intcom](http://www.elsevier.com/locate/intcom)

# Using handhelds for wireless remote control of PCs and appliances

Brad A. Myers<sup>\*,1</sup>

*Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University,  
Pittsburgh, PA 15213-3891, USA*

Received 16 January 2004; revised 31 May 2004; accepted 11 June 2004

Available online 28 July 2004

---

## Abstract

This article provides an overview of the capabilities that we are developing as part of the Pebbles research project for wireless handheld devices such as mobile phones and palm-size computers like Palm Organizers and PocketPCs. Instead of just being used as a phone or organizer, handheld devices can also be used as remote controls for computers and household and office appliances.

© 2004 Elsevier B.V. All rights reserved.

*Keywords:* Pebbles; Handhelds; Personal digital assistants; Remote control; Appliances

---

## 1. Introduction

We believe that handhelds can improve the user interfaces of many other devices, rather than just being another gadget to be learned. Imagine the following scenario:

You come home and aim your Smartphone at your garage, and push a button on the phone. The garage door opens. As you enter, your phone displays a diagram of the lights and appliances in your home, and you tap on the entryway light to turn it on. When you walk into the family room, the phone display changes and shows various commands useful for the entertainment system. You hit 'Play DVD', and the phone turns the DVD player on, switches the TV to INPUT-3 where the DVD is connected,

---

\* Tel.: +1-412-268-5150; fax: +1-412-268-1266.

E-mail address: [bam+@cs.cmu.edu](mailto:bam+@cs.cmu.edu) (B.A. Myers).

<sup>1</sup> <http://www.cs.cmu.edu/~bam>.

turns on the stereo and switches it to AUX input, and finally, the starts the DVD playing. Later, you go to your home office and put the phone in its recharging cradle next to the computer. The phone display changes to serve as an extra screen for the application that is currently in use on the PC. When browsing the web, for example, the phone display has big BACK and FORWARD buttons, as well as a scroll bar for swiftly moving through pages. Since the phone is conveniently located to the left of the keyboard, you can tap on the phone display without looking to scroll and change pages, while using the mouse in your right hand. In the evening, you use the Smartphone as the remote control for your bedroom television, and also to set the alarm on the clock beside the bed. Later, you get a call from a colleague to say that a meeting is delayed for an hour, so you can sleep later than expected. Changing the alarm time automatically adjusts the thermostat to keep the household temperatures at the nighttime setting for a little longer, thereby saving some energy. The coffee maker is also automatically delayed an hour. In the morning, when you enter your car, you put the phone into its cradle, and it sends the meeting location that you had been emailed to the car's navigation system. After arriving at your destination, you give your presentation using the Smartphone display as a remote control for the slide show. You see your notes and a thumbnail-size picture of your slide on the phone's display, and you write on the phone's screen to draw on your slides. Pressing buttons on your phone with your thumb allows you to easily move back and forward in the presentation and switch to demonstrations and back to the slide show without fumbling.

All of these ideas are being investigated by the Pebbles research project at Carnegie Mellon University. They can be demonstrated now, and may soon be available in commercial products. This article summarizes the Pebbles project, focusing on the applications we have created to allow handheld devices to be used as remote controls for applications running on PCs, and for everyday appliances.

## 2. What is a 'handheld' anyway?

What exactly is a 'handheld device'? I define a handheld device as *a computerized, electronic machine that is designed to be held in one hand*. The definition clearly includes calculators, organizers, pagers, mobile phones (generally called 'cell phones' in the US), and Personal Digital Assistants (PDAs) such as the Newton, Palm and PocketPC. All of these PDAs are designed to fit into one hand, and have a touch-sensitive screen on which a stylus can write. The built-in functions include a calendar, address book, a 'to-do' list, and memo pad for taking notes. These devices are programmable, and it is relatively easy to add other applications that can be downloaded from the Internet.

There are about 30 million PDAs in the world, but this pales in comparison to the 1.3 billion mobile phone devices worldwide (European Cellular Network, 2003). Increasingly, these mobile phones contain PDA-like capabilities, and are then often classified as 'Smartphones'. The sales of Smartphones were predicted to be 4 million units in Europe in 2003, beating the sales of PDAs, according to Canlys.com (RIMRoad News, 2003).

Clearly, handhelds are becoming devices used by an increasingly large percentage of the world's population.

What is *not* included by the term handheld? My definition excludes laptop computers, for example, since they cannot be easily used while being held in one hand; laptops are designed to be used while sitting, with the computer on a table or in your lap. Also excluded are so-called 'wearable' devices, such as special eye-glasses or heads-up displays, since these are not designed to be held. However, many of the applications and user interface issues discussed in this article could also apply to such wearable devices.

It is less clear whether or not to classify devices such as TabletPCs and 'clamshell' Windows CE devices as handhelds. TabletPCs run a full-functioning operating system (e.g. Windows XP), but are designed to be used like a writing tablet rather than with a keyboard (Fig. 1a). These might alternatively be called 'arm-helds' because they are too big to be held in one hand without using an arm. Another class of devices that are questionably called "handheld" is represented by the horizontal Windows CE devices such as the Jornada 680, which have built-in keyboards (Fig. 1b). Although small, these devices are very awkward to use while being held in one hand, and usually must be placed on a horizontal surface. Therefore, I do not call tablet computers or clamshell Windows CE devices "handhelds".

### 3. What makes this scenario possible now?

How can the opening scenario be possible? There are a large number of technologies in development today that will soon be ready for widespread use that will make scenarios like the one above possible. These can be broken into advances with handhelds, with communication, and with appliances.

#### 3.1. *Advances with handhelds*

Handheld devices are getting more powerful. Today's PDAs often run at 400 MHz, which is as fast as the PCs of just 4 years ago. In fact, the speed of the processors for handhelds, and the size of their memories, is following the well-known Moore's law for computers; doubling about every year and a half. Therefore, almost any application that could be imagined running on a PC will find adequate performance on a handheld device.

Processors in mobile phones are also getting faster. Phone manufacturers are adding more functions and capabilities to phones, and most mobile phones today are capable of browsing the Internet and running a Java virtual machine. Manufacturers are pushing towards so-called Smartphones for which a variety of applications can be downloaded, just like for PDAs. Some Smartphones provide PalmOS or Windows CE operating systems and user interfaces, though such devices usually have a larger form-factor than conventional mobile phones. Other devices run operating systems specially designed for mobile phones, such as Symbian. Newer phones also include cameras, voice recognition, touch screens, and other technologies.

The displays on the Newton and the first Palms were black and white, but current versions of all PDAs increasingly use back-lit color screens, which are much easier to read



Fig. 1. TabletPC (a) and HPC WindowsCE device (b).

in most lighting conditions. Back-lit screens can be harder to read in full sunlight, but many devices, such as the Compaq iPaqs, use side-lit color screens, which are readable even in bright light. Mobile phones were once limited to small five-line displays, but now increasingly have larger color displays. These displays are often smaller than PDA screens, however, since people prefer smaller phone devices.

Battery life continues to improve, with color-display devices lasting at least 1 or 2 days between recharging. However, color devices with back or side lights still do not get the long life of the older black-and-white devices.

### 3.2. *Advances with communication technology*

In the original vision for ubiquitous computing (Weiser, 1993), all the devices would be in continuous communication with each other. The original Xerox PARCTabs (Want et al., 1995) used a custom infrared (IR) network to stay connected to the rest of the computers in the environment. However, the first generation of commercial PDAs did not have any communication abilities. For example, Sharp ‘organizers’ often had tiny keyboards and a number of handheld functions, but did not communicate with PCs at all. The first model of the Apple Newton only provided connectivity with other computers as an extra-cost option. One reason for the great success of the first Palm, released in 1996, was that it could easily synchronize all of its data with a desktop computer using a one-button HotSync™. PalmOS devices had built-in infrared wireless communication starting about 1998, which allowed Palms to ‘beam’ information to each other. Limitations of IR include that the handheld must be carefully aimed at the receiver, and the IR in handhelds tend to be very short ranged. Often the sending and receiving devices need to be less than 2 ft apart. This makes communicating using IR inappropriate for most of the scenarios described in this article, where the handheld may be at some distance from the device to be controlled, and may not be pointing at it.

Meanwhile, laptops were starting to get access to wireless technologies such as 802.11, which first appeared around 1994, but did not become widespread until around 2000. The most popular version is 802.11b, which is now also called ‘Wi-Fi’. Initially, getting Wi-Fi required using a PC card (also called PCMCIA) for a laptop. Few of the early handheld devices could accept a PC card, and none had driver support for Wi-Fi cards until the Compaq iPaq, in about 2000. Eventually, handhelds with built-in Wi-Fi appeared, and smaller Wi-Fi cards (such as the CompactFlash form-factor) allowed Wi-Fi to be used with more handheld devices. Now, it is possible to get Wi-Fi access on many different kinds of PDAs. A problem with Wi-Fi, however, continues to be its high power usage. Using Wi-Fi communication on a current iPaq 5455 drains the battery in less than an hour.

Other radio technologies have addressed the power problem. In addition to research systems (Shih et al., 2002), the Bluetooth™ radio network technology was designed from the beginning to have low power usage. Bluetooth research started in 1994, but the standard was not released until 1998 with the technology not becoming widespread until 2003. Handheld devices with built-in Bluetooth are now available, and are becoming particularly common in the mobile phone market. Unlike Wi-Fi, which connects devices to the internet, Bluetooth is used primarily for connecting one device to one other device—such as a handheld to a personal computer—which is all that is required to

support most of the concepts described in this article. BlueTooth includes techniques for devices to find each other (also called ‘Device Discovery’), which is important for the scenarios we are investigating, but which is not an area we are currently investigating (Section 5). Unfortunately, the device discovery and set-up time for BlueTooth can take 5–30 s, which means that BlueTooth is more appropriate for environments where new devices only appear rarely.

Another wireless technology is the mobile phone network. The mobile network is increasingly able to carry data, and therefore is relevant to handhelds interfacing with other technology. Currently, in the USA, it is easy to get data rates at 19.2 kHz, with some phone companies offering about 100 kHz with specialized interface cards. In Europe and Japan, higher data rates are already available, and will eventually be available in the USA. For the applications described here, even the low data rates may be sufficient. The main criterion is often not *bandwidth*—the number of bits-per-second. Instead, the main issue is *latency*—or how long it takes a message to get from the sender to the receiver. For a remote control device to say ‘turn on’ may only take a few bytes, but the receiver must be able to receive that message quickly, ideally in less than one-half-a-second, so that the user feels that the device is responsive. Unfortunately, most mobile networks today are optimized for bandwidth, and can take up to 30 s or even longer to transmit a message. If a number has to be dialed first and a connection made, then sending a message can take over a minute. Therefore, today’s mobile technology (at least in the USA) does not seem appropriate for the kinds of applications described here. However, we expect that mobile phone data communication with low-latency and zero connect time will be available in the coming years.

There are many other wireless technologies also being researched. It seems clear that *some* kind of wireless technology will be available that will allow current and future handheld devices to communicate whenever necessary with computers and other devices in their vicinity. Furthermore, with the plethora of different communication technologies that will be available, people will *expect* their devices to communicate with each other.

### 3.3. Advances with appliances

Meanwhile, appliances are also evolving to be able to communicate with other devices. The ubiquitous one-way infrared remote allows a device to control an appliance, but this does not allow the device to *query* the appliance to find out its status, which is necessary for many of the scenarios envisioned here. However, it would be possible to connect with today’s existing appliances using IR.

More exciting are new developments that hold promise for *two-way* communication between devices and appliances. A number of rudimentary protocols exist today for commercial appliances, and standard bodies are working on many more.

For example, some Sony audio-visual appliances support a protocol called ‘S-Link’ that enables one Sony appliance to communicate with and control another appliance. Although it is a closed, proprietary protocol only supported by Sony appliances, it has been adapted to allow computers to control Sony devices.<sup>2</sup>

<sup>2</sup> See, for example, information at <http://www.brian-patti.com/s-link/>.

Digital video devices, such as camcorders and high-end VCRs, often provide video output on a FireWire cable, which is also called IEEE 1394 (the official name of the standard), or i.Link. Many computers also have FireWire ports. FireWire supports two-way protocols, which allow the receiver, such as a computer, to send commands back to the sender, such as the camcorder, to tell it to play, stop, pause, and so on. Currently, we can remotely control a Sony DV camcorder using the AV/C protocol through FireWire from a Windows XP computer. Our remote-control device can then communicate with the computer using 802.11 or BlueTooth, and the data will be forwarded to the camcorder through the FireWire cable.

More promising is a large standardization effort for UPnP (which stands for *Universal Plug and Play*), which is supported by over 600 companies. The goal of UPnP is to ‘create the means to easily connect devices and simplify the implementation of networks... UPnP technology is all about making home networking simple and affordable for users so the connected home experience becomes a mainstream experience for users and a great opportunity for the industry’ (<http://www.upnp.org>). UPnP provides standard protocols for controlling appliances and receiving feedback about state, and defines standard sets of functionality for different classes of devices. Already, there are standards for devices such as printers, audio-visual equipment, lighting, and HVAC (heating, venting and air conditioning) equipment, and many other devices are being standardized. A number of appliances supporting UPnP are starting to appear, and others have been announced or planned.

Meanwhile, a number of industry groups have been formed to study or promote the ‘connected appliances’ or the ‘internet-ready home’. For example, the Internet Home Alliance is a ‘cross-industry network of leading companies advancing the home technology market ... (and) to accelerate the development of the market for home technologies that require a broadband or persistent connection to the Internet’ (<http://www.internethomealliance.com/>).

The result of all of these initiatives will be more and more appliances that can talk to computers, and therefore will support at least some of the capabilities that we would need for the scenarios discussed here. If the appliances cannot directly support the remote control from handhelds, then it might still be possible by using a computer as an intermediary (which we call a ‘bridge’ or ‘adapter’). Eventually, we hope the kinds of capabilities we and others are demonstrating will convince appliance and handheld manufacturers to build these technologies into future devices.

#### **4. Overview of the Pebbles project**

The Pebbles project (Myers, 2001) (<http://www.pebbles.hcii.cmu.edu>) is investigating the many ways that handheld devices can be used at the same time as other computerized devices. Pebbles stands for *P*DAs for the *E*ntry of *B*oth *B*ytes and *L*ocations from *E*xternal *S*ources.

#### 4.1. Remote control of PCs

One aspect of this work is the remote control of PCs from handheld devices. We have created a wide range of applications that allow a handheld device to provide input and serve as the output to control applications running on a PC.

For example, in design reviews, brainstorming sessions, and organizational meetings, a PC is often used to display slides or a current plan, and the people in attendance provide input. Our *RemoteCommander* application (Fig. 2a,b) allows each person to use their PDA to control the PC's cursor and keyboard input from their seat (Myers et al., 1998). This will allow each person to participate without having to jump up and grab the PC's one mouse and keyboard. In the PalmOS version (Fig. 2a), the PDA provides mouse movement like a touchpad on a laptop, and the hard or soft buttons can be used for the mouse buttons. Graffiti or a soft keyboard can be used for typing, and word prediction is included. In addition to these functions, the PocketPC version (Fig. 2b) also downloads a picture of

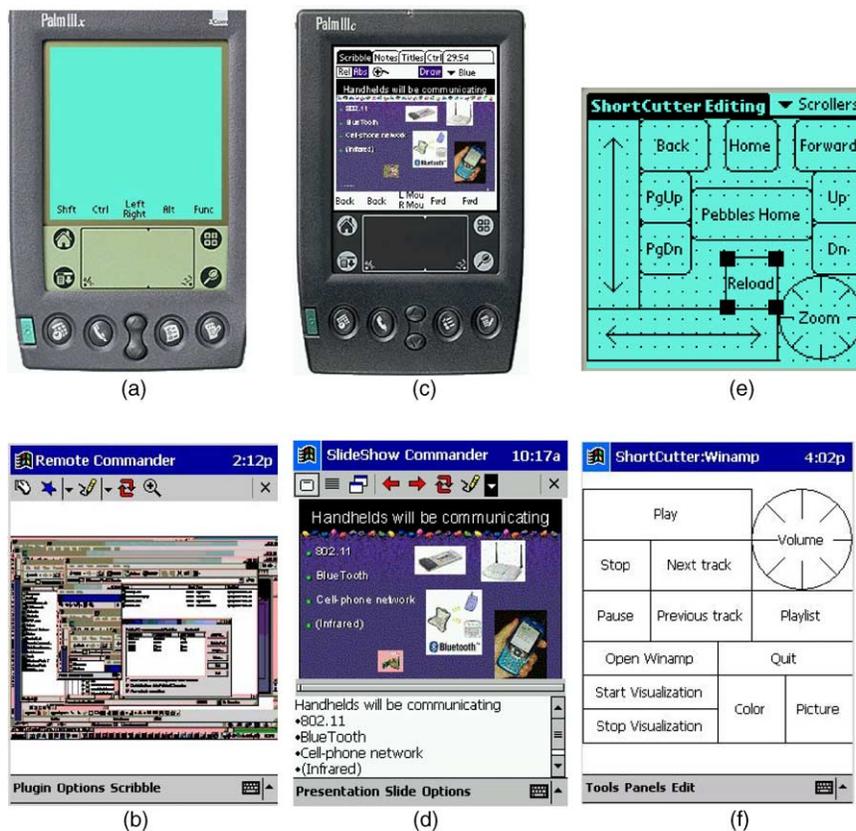


Fig. 2. RemoteCommander on a Palm (a) and a PocketPC (b). The PocketPC has sufficient bandwidth to display a screenshot of the PC. SlideShow Commander on a Palm (c) and PocketPC (d). Shortcutter in edit mode on a Palm, where the user is creating a scrolling panel (e). Shortcutter in run mode on a PocketPC with a panel for controlling the WinAmp media player (f).

the PC screen to the PDA, which can be zoomed and scrolled to see different parts of the PC's screen.

We also discovered that RemoteCommander was valuable as a replacement for the keyboard and mouse for people with certain physical disabilities, including Muscular Dystrophy and some forms of Cerebral Palsy (Myers et al., 2002b). People with these disabilities have difficulty moving their arms and hands, but may retain good control of their fingers, which often makes using a stylus for cursor control and typing using tiny keyboard easier than with standard input devices. We are now looking into other ways that handhelds might help people with disabilities control their computers, including addressing the problem of text input for people with motor control difficulties (Wobbrock et al., 2003).

Our *SlideShow Commander* application (Fig. 2c,d) allows a PDA to remotely control the PC when running a PowerPoint presentation. On the PDA, the user can see a small picture and the notes for the current slide, and can advance the slide show to the next or previous slide or any slide chosen from the list of slide titles. The user can also draw on the current slide by drawing on the PDA, and can easily switch to and from other applications running on the PC, for example for a live demonstration. SlideShow Commander helps make presentations proceed more smoothly.

In other research, we are investigating how a PDA can be useful in augmenting the Windows user interface for desktop applications (Myers et al., 2000b). For example, our *Shortcutter* application (Fig. 2e,f) allows panels of controls to be drawn by direct manipulation on the PDA, and then used to control any PC application. The user can choose from various widgets on the PDA, including buttons, sliders and a virtual knob, and assign actions to the widgets, including sending any keyboard key to the PC, running an application on the PC, scrolling, mouse buttons, or a macro containing multiple commands. One way people have used Shortcutter is to remotely control media players on PCs, like the Windows Media Player or WinAmp (Fig. 2f). Another use is to put scrolling controls on the PDA, and then use the PDA in a cradle on the non-dominant side of the keyboard, with the mouse on the other side. We performed a user study that showed that the PDA could be used effectively in the left hand as a scrolling device for desktop applications (Myers et al., 2000a). In our study, we found the cradle that comes with the Palm to be too unstable for use in this manner, so we instead put the PDA flat on the table and used a cable, such as provided with some PDAs like the Palm m100 and Tungsten E.

#### 4.2. Remote control of appliances

Increasingly, home and office appliances, including televisions, VCRs, stereo equipment, ovens, thermostats, light switches, telephones, and factory equipment, are designed with many complex functions, and often come with remote controls. However, the trend has been that as appliances get more computerized with more features, their user interfaces become harder to use (Brouwer-Janse et al., 1992). Our approach to address this problem is to move the user interface onto a handheld, where increased processing power and better input-output capabilities can be used to improve the usability. Since it will be a *personal* device, consistency can be provided, so that whenever the user needs to perform a function, such as setting the time, it will always be done the same way.

We call this approach the ‘Personal Universal Controller’ (PUC). We are now developing *automatic* user interface generators that will take a high-level, abstract specification of the functions of an appliance, and create a high-quality user interface for it (Fig. 3). We have created generators that will automatically produce graphical user interface control panels for PDAs, for desktop computers, and also for Smartphones that do not have touch screens (Nichols et al., 2002). Another generator uses the same specification, and automatically creates speech interfaces using the *Universal Speech Interface* framework (Rosenfeld et al., 2001). Multiple PUCs can be communicating with the same or different appliances at the same time, which enables multi-modal control of appliances, where the user can speak some commands and use a PDA for others.

We have created remote control interfaces for a variety of real and simulated appliances to demonstrate the range of the PUC approach. Real appliances, such as the Lutron Home Lighting System, the Axis Pan and Tilt Camera, and a Sony Camcorder, can be addressed through standard and proprietary protocols, including UPnP, AV/C and X10. Simulated devices help show the range of our specification language beyond the appliances that we can currently control. We have simulated interfaces for an elevator, and the navigation, heating, and driver information center of a GMC Denali sport utility vehicle (SUV).

An important issue will be whether users will accept this form of remote control. In a preliminary study, we designed interfaces by hand for a PDA to remote control an AT&T Telephone/Answering machine, and an AIWA shelf stereo. We found that users were twice as fast and made half as many errors when using our prototype interfaces as compared to the manufacturers’ interfaces (Nichols and Myers, 2003). We also observed remote control use in homes, and observed that a few physical buttons seem to be often used without looking (e.g. volume and channel), so it may be useful to provide these as physical buttons on the PDA, while the rest of the controls are on the screen. Fortunately, all PDAs and mobile phones have a few physical buttons that could be used for these purposes.

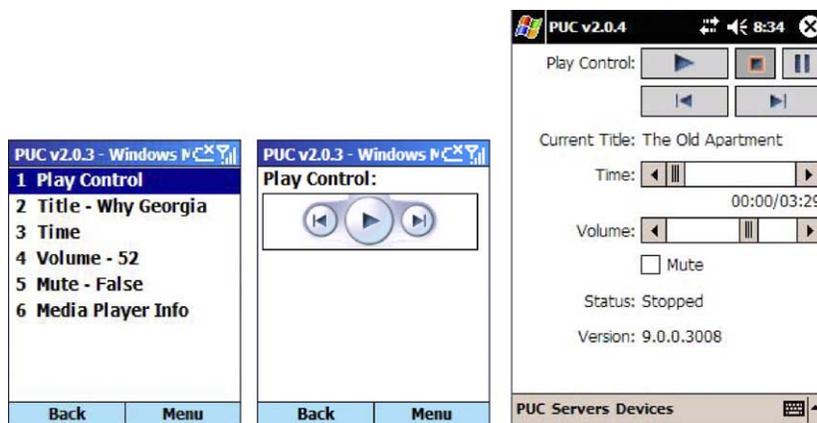


Fig. 3. User interfaces which were automatically generated by PUC for controlling Windows Media Player. On the left, two screens from a Smartphone interface. On the right, a screen from a PocketPC interface (Nichols et al., 2004).

## 5. Related work

A few other projects have looked at the general issues with PDAs interacting with stationary devices, including the original Xerox PARCTab (Want et al., 1995), and SharedNotes implemented with GroupKit (Greenberg et al., 1999).

Rekimoto has developed many systems that explore connecting multiple devices, including the ‘Pick-and-Drop’ interaction technique for transferring information (Rekimoto, 1997), a multi-device drawing tool (Rekimoto, 1998), and ‘HyperDragging’ to move information from one device’s display to another (Rekimoto and Saitoh, 1999). The Pebbles approach augments this research by providing a framework and architecture to support new applications and new devices that can interoperate.

A number of research groups are working on controlling appliances from handheld devices, but Pebbles is the only project that automatically creates high-quality interfaces without user intervention. Hodes proposed a ‘universal interactor’ that can adapt itself to control many devices (Hodes et al., 1997). Unlike the PUC, that research focuses on the system and infrastructure issues rather than how to create user interfaces. An IBM project (Eustice et al., 1999) describes a ‘Universal Information Appliance’ (UIA) that might be implemented on a PDA. The UIA uses an XML-based language called MoDAL from which it creates a user interface panel for accessing information. However, the MoDAL processor apparently only handles simple layouts, and its only type of input control is text strings. The Stanford ICrafter (Ponnekanti et al., 2001) is a framework for distributing appliance interfaces to many different controlling devices. While their framework supports the automatic generation of interfaces, their paper focuses on hand-generated interfaces and shows only one simple automatically generated interface. They also mention the difficulty of generating speech interfaces.

The XWeb project (Olsen et al., 2000) is working to separate the functionality of the appliance from the device upon which it is displayed. XWeb defines an XML language from which user interfaces can be created. Unlike the PUC specification language, XWeb’s language uses only a tree for specifying structural information about an appliance. Their approach seems to work well for interfaces that have no modes, but it is unclear how well it would work for remote control interfaces, where modes are commonplace. XWeb also supports the construction of speech interfaces. Their approach to speech interface design, including emphasis on a fixed language and cross-application skill transference, is quite similar to ours, as it is derived from a joint philosophy (Rosenfeld et al., 2001). XWeb’s language design allows users to directly traverse and manipulate tree structures by speech. They report, however, that this is a hard concept for users to grasp (Olsen et al., 2000). CMU’s Universal Speech Interface design differs by trying to stay closer to the way people might talk about the task itself, and is somewhat closer to naturally generated speech.

The INCITS V2 standardization effort (Zimmermann et al., 2002) is developing the Alternative Interface Access Protocol (AIAP) to help disabled people use everyday appliances with an approach similar to the PUC. AIAP contains a description language for appliances that different interface generators use to create interfaces for both common devices, like the PocketPC, and specialized devices, such as an interactive Braille pad. We have begun collaborating with the V2 group and plan to continue to do so in the future.

One area we are not addressing is how the devices find each other, which is also called ‘Device Discovery’. In Pebbles, we just select the desired device from a list or type in its address. BlueTooth has some device discovery capabilities, but they are slow and awkward. The early PARCTab explored ‘proximate selection’ where devices in the vicinity could be more easily selected (Schilit et al., 1994). With ‘SyncTap’, tapping on both devices simultaneously causes a connection (Rekimoto et al., 2003). Our approach of deferring the discovery process to lower-level protocols is common to many other research systems (Edwards et al., 2002).

## 6. Conclusion

The research on the Pebbles project is on-going. This article has shown the context for current and future work, as well as our current status. The software we have developed as part of the Pebbles project is mostly available for free download from our web site, <http://www.pebbles.hcii.cmu.edu/>. One Pebbles application, the SlideShow Commander, was licensed for commercial sale (available from <http://www.handango.com>). In the future, we will be working on improving the interfaces to these applications with a particular focus on improving handheld usability for people with different kinds of disabilities. Other work will be directed at improving the remote control capabilities for appliances, especially for *collections* of appliances, so the user can, for example, control an entire entertainment system with a single command (such as Play DVD), instead of having to control each component separately.

One area we started to explore is public-private data sharing. In the ‘Command Post of the Future’ project, we used handhelds to ‘drill down’ and get details about publicly displayed information (Myers et al., 2002a). This area has been explored by others as well (Rekimoto, 1998; Greenberg et al., 1999), but much more work could be done.

As appliances and handhelds progress towards increased functionality and increased ability to communicate, people will expect these kinds of capabilities to be provided. By using the handheld as a remote control for other devices, all of the user’s information and control can have a consistent user interface and an integrated information space. This should ease the total burden on users of having lots of appliances while at the same time providing increased functionality.

## Acknowledgements

The Pebbles research project has been made possible by the efforts of over 30 students. I especially want to thank Rob Miller, Jeff Nichols, and Jake Wobbrock. This work was funded in part by grants from DARPA, NSF, Microsoft, General Motors, NEC Foundation of America, and the Pittsburgh Digital Greenhouse, and equipment grants from Mitsubishi Electric Research Laboratories, VividLogic, Symbol Technologies, Hewlett-Packard, Palm, IBM, Smart Technologies, TDK, Lantronix, and Lucent. For help with this paper, I would like to thank Jeff Nichols, Jacob Wobbrock, Htet Htet Aung, and Andrew Faulring.

## References

- Brouwer-Janse, M.D., Bennett, R.W., Endo, T., van Nes, F.L., Strubbe, H.J., Gentner, D.R., 1992. Interfaces for consumer products: 'how to camouflage the computer?'. CHI'1992: Human Factors in Computing Systems, Monterey, CA, May 3–7, pp. 287–290.
- Edwards, W.K., et al., 2002. Using speakeasy for ad hoc peer-to-peer collaboration. ACM Conference on Computer Supported Cooperative Work: CSCW'02, New Orleans, LA, pp. 256–265.
- European Cellular Network, 2003. Stats Snapshot 5/2003 June, <http://www.cellular.co.za/stats/stats-main.htm>.
- Eustice, K.F., Lehman, T.J., Morales, A., Munson, M.C., Edlund, S., Guillen, M., 1999. A universal information appliance. IBM Systems Journal 38(4), 575–601. <http://www.research.ibm.com/journal/sj/384/eustice.html>.
- Greenberg, S., Boyle, M., Laberg, J., 1999. PDAs and shared public displays: making personal information public, and public information personal. Personal Technologies 3(1), 54–64.
- Hodes, T.D., Katz, R.H., Servan-Schreiber, E., Rowe, L., 1997. Composable ad-hoc mobile services for universal interaction. Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking (ACM Mobicom'97), Budapest Hungary, September 26–30, pp. 1–12.
- Myers, B.A., Stiel, H., Gargiulo, R., 1998. Collaboration using multiple PDAs connected to a PC. Proceedings CSCW'98: ACM Conference on Computer-Supported Cooperative Work, Seattle, WA, November 14–18, pp. 285–294, <http://www.cs.cmu.edu/~pebbles>.
- Myers, B.A., (Leo) Lie, K.P., (Jerry) Yang, B.-C., 2000a. Two-handed input using a PDA and a mouse. Human Factors in Computing Systems, Proceedings CHI'2000, The Hague, The Netherlands, April 1–6, pp. 41–48.
- Myers, B.A., Miller, R.C., Bostwick, B., Evankovich, C., 2000b. Extending the windows desktop interface with connected handheld computers. Fourth USENIX Windows Systems Symposium, Seattle, WA, August 3–4, pp. 79–88.
- Myers, B.A., 2001. Using hand-held devices and PCs together. Communications of the ACM 4(11), 34–41. <http://www.cs.cmu.edu/~pebbles/papers/pebblescacm.pdf>.
- Myers, B.A., et al., 2002a. Flexi-modal and multi-machine user interfaces. IEEE Fourth International Conference on Multimodal Interfaces, Pittsburgh, PA, October 14–16, pp. 343–348, <http://www.cs.cmu.edu/~cpof/papers/cpoficmi02.pdf>.
- Myers, B.A., Wobbrock, J.O., Yang, S., Yeung, B., Nichols, J., Miller, R., 2002b. Using handhelds to help people with motor impairments. Fifth International ACM SIGCAPH Conference on Assistive Technologies: ASSETS'02, Scotland, pp. 89–96.
- Nichols, J., Myers, B.A., 2003. Studying the use of handhelds to control smart appliances. International Workshop on Smart Appliances and Wearable Computers, Providence, Rhode Island, May 19–22, pp. 274–279, <http://www.cs.cmu.edu/~effreyn/papers/iwsawc2003puc.pdf>.
- Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., Pignol, M., 2002. Generating remote control interfaces for complex appliances. CHI Letters: ACM Symposium on User Interface Software and Technology, UIST'02, Paris, France, October 2002, pp. 161–170, <http://www.cs.cmu.edu/~pebbles/papers/PebblesPUCuist.pdf>.
- Nichols, J., Myers, B.A., Litwack, K., 2004. Improving Automatic Interface Generation with Smart Templates, IUI'04, Madeira, Funchal, Portugal, January 13–16, 286–288, <http://www.cs.cmu.edu/~jeffreyn/papers/templatesIUI.pdf>.
- Olsen, D.R. Jr., Jefferies, S., Nielsen, T., Moyes, W., Fredrickson, P., 2000. Cross-modal Interaction using Xweb. Proceedings UIST'00: ACM SIGGRAPH Symposium on User Interface Software and Technology, San Diego, CA, pp. 191–200.
- Ponnekanti, S.R., Lee, B., Fox, A., Hanrahan, P., Winograd, T., 2001. ICrafter: A Service Framework for Ubiquitous Computing Environments, UBICOMP, Atlanta, Georgia, pp. 56–75.
- Rekimoto, J., 1997. Pick-and-drop: a direct manipulation technique for multiple computer environments. ACM SIGGRAPH Symposium on User Interface Software and Technology, Proceedings UIST'97. Banff, Alberta, Canada, October 14–17, pp. 31–39.
- Rekimoto, J., 1998. A multiple device approach for supporting whiteboard-based interactions. Human Factors in Computing Systems, Proceedings SIGCHI'98, Los Angeles, CA, April, pp. 344–351.

- Rekimoto, J., Saitoh, M., 1999. Augmented surfaces: a spatially continuous work space for hybrid computing environments. *Human Factors in Computing Systems, Proceedings SIGCHI'99*, Pittsburgh, PA, May, pp. 378–385.
- Rekimoto, J., Ayatsuka, Y., Kohno, M., 2003. SyncTap: an interaction technique for mobile networking. *Mobile HCI*. Springer, Berlin, pp. 104–115.
- RIMRoad News, 2003. Smartphones to Overtake Handhelds This Year, <http://www.rimroad.com/articles/2003/3/2003-3-27-Smartphones-to-Overtake.html>.
- Rosenfeld, R., Olsen, D., Rudnicky, A., 2001. Universal speech interfaces. *ACM Interactions* VIII(6), 34–44.
- Schilit, B.N., Adams, N.I., Want, R., 1994. Context-aware computing applications. *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA. IEEE Computer Society, pp. 85–90.
- Shih, E., Bahl, P., Sinclair, M.J., 2002. Wake on wireless: an event driven energy saving strategy for battery operated devices. *MOBICOM'02: Eighth Annual ACM International Conference on Mobile Computing and Networking*, Atlanta, GA, pp. 160–171.
- Want, R., et al., 1995. An overview of the ParcTab ubiquitous computing experiment. *IEEE Personal Communications*, 28–43. Also appears as Xerox PARC Technical Report CSL-95-1, March, 1995.
- Weiser, M., 1993. Some computer science issues in ubiquitous computing. *CACM* 36(7), 74–83.
- Wobbrock, J.O., Myers, B.A., Kembel, J., 2003. EdgeWrite: a high-accuracy stylus text entry method. *CHI Letters: ACM Symposium on User Interface Software and Technology, UIST'03*, Vancouver, British Columbia, Canada, November 2–5, pp. 61–70, <http://www.cs.cmu.edu/~jrock/pubs/uist-03.pdf>.
- Zimmermann, G., Vanderheiden, G., Gilman, A., 2002. Prototype implementations for a universal remote console specification. *CHI'2002*, Minneapolis, MN, pp. 510–511.